AD-A096 693 GEORGE WASHINGTON UNIV WASHINGTON DC INST FOR MANAGE--ETC F/G 9/2
A USER'S MANUAL FOR SENSUMT: A PENALTY FUNCTION COMPUTER PROGRA--ETC(U)
OCT 80 A V FIACCO, A GHAEMI DAAG29-79-C-0062
UNCLASSIFIED SERIAL-T-434 ARO-16229.9-M NL

1 OF 2
AD A 096693

LEVEL II

12

THE
GEORGE
WASHINGTON
UNIVERSITY

STUDENTS FACULTY STUDY R ESEARCH DEVELOPMENT FUT URE CAREER CREATIVITY CC MMUNITY LEADERSHIP TECH NOLOGY FRONTIE ESIGN ENGINEERING APP ENC GEORGE WASHIN NIV

SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

(6)

A USER'S MANUAL FOR SENSUMT:

A Penalty Function Computer Program for Solution, Sensitivity Analysis, and Optimal Value Bound Calculation in Parametric Nonlinear Programs

(10) by

Anthony V. Fiacco
Abolfazl Ghaemi

(9) Scientific Rept. 2

(12) AIRO

(19) 16227.7-M

(14) Serial T-434
~~21 October~~ 1980

(11) 21 Oct 80

(13) 124

The George Washington University
School of Engineering and Applied Science
Institute for Management Science and Engineering

(15)

Contract DAAG 29-79-C-0062,
US Army Research Office-Durham

Contract N00014-75-C-0729
Office of Naval Research

Grant ENG-7906104
National Science Foundation

406743

NONE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER T-434	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)* A USER'S MANUAL FOR SENSUMT: A PENALTY FUNCTION COMPUTER PROGRAM FOR SOLUTION, SENSITIVITY ANALYSIS, AND OPTIMAL VALUE BOUND CALCULATION IN PARAMETRIC NONLINEAR PROGRAMS	5. TYPE OF REPORT & PERIOD COVERED: SCIENTIFIC 6. PERFORMING ORG. REPORT NUMBER: T-434
7. AUTHOR(s): ANTHONY V. FIACCO, ABOLFAZL GHAEMI	8. CONTRACT OR GRANT NUMBER(s): DAAG 29-79-C0062, N00014-75-C-0729, ENG-7906104
9. PERFORMING ORGANIZATION NAME AND ADDRESS: THE GEORGE WASHINGTON UNIVERSITY, INSTITUTE FOR MANAGEMENT SCIENCE & ENGINEERING, WASHINGTON, DC 20052	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Research Office, Box 12211, Research Triangle Park, NC 27709; Office of Naval Research, Code 434, Arlington, VA 22217	12. REPORT DATE: 21 October 1980 13. NUMBER OF PAGES: 117
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: National Science Foundation, Division of Electrical, Computer, and Sys. Engr., Systems Theory and Operations Research, Washington, DC 20550	15. SECURITY CLASS. *(of this report)*: NONE 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

APPROVED FOR PUBLIC SALE AND RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

NONLINEAR PARAMETRIC PROGRAMMING
SENSUMT
SENSITIVITY
PARAMETRIC BOUNDS
OPTIMAL VALUE FUNCTION
OPTIMAL SOLUTION

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This manual is intended to serve as a guide for coding, solving, and conducting sensitivity and optimal value bound analysis for parametric nonlinear programming problems with the computer programming model SENSUMT. The basic sensitivity results and bound calculation techniques are briefly reviewed and the algorithms implementing them are presented. A procedure for coding problems for SENSUMT and detailed illustrations of the computer

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601 |

NONE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. Abstract (continued)

output is included. For completeness, the computer listing and a brief description of all the subroutines comprising SENSUMT, many of which are taken intact from a previously developed program SUMT-Version 4, are also provided.

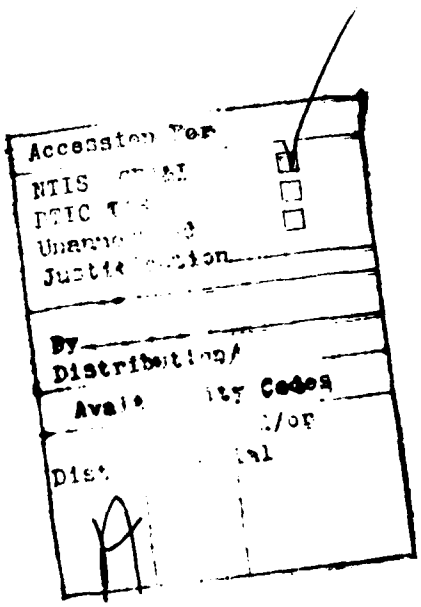


TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Abstract
of
Serial T-434
21 October 1980

A USER'S MANUAL FOR SENSUMT:

A Penalty Function Computer Program for Solution, Sensitivity Analysis, and Optimal Value Bound Calculation in Parametric Nonlinear Programs

by

Anthony V. Fiacco
Abolfazl Ghaemi

This manual is intended to serve as a guide for coding, solving, and conducting sensitivity and optimal value bound analysis for parametric nonlinear programming problems with the computer programming model SENSUMT. The basic sensitivity results and bound calculation techniques are briefly reviewed and the algorithms implementing them are presented. A procedure for coding problems for SENSUMT and detailed illustrations of the computer output is included. For completeness, the computer listing and a brief description of all the subroutines comprising SENSUMT, many of which are taken intact from a previously developed program SUMT-Version 4, are also provided.

Research Supported by

Contract DAAG 29-79-C-0062
US Army Research Office—Durham

Contract N00014-75-C-0729
Office of Naval Research

Grant ENG-7906104
National Science Foundation

ACKNOWLEDGMENT

This latest version of the SENSUMT program is an experimental program that is still evolving, though it has been used routinely at The George Washington University over the past several years to solve intermediate sized (up to about 100 variables and constraints) NLP problems and to conduct solution sensitivity analysis. A major part of it is the product of many theoretical, computational and software developments contributed by others. Several subroutines have been taken essentially intact from existing codes, most notably from SUMT-Version 4, coded by W. C. Mylander, R. L. Holmes, and G. P. McCormick; the first version of SENSUMT coded by R. L. Armacost and W. C. Mylander; and an extended version of SENSUMT coded subsequently by Armacost. The authors wish especially to acknowledge these contributions, along with the first program to implement some of the techniques used by this general approach for the calculation of sensitivity analysis, coded by B. Causey.

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

A USER'S MANUAL FOR SENSUMT:

A Penalty Function Computer Program for Solution, Sensitivity Analysis, and Optimal Value Bound Calculation in Parametric Nonlinear Programs

by

Anthony V. Fiacco
Abolfazl Ghaemi

1. Introduction

This is a manual intended to facilitate the use of the computer program SENSUMT. This program is capable of solving general parametric nonlinear programs and conducting a sensitivity analysis using the Sequential Unconstrained Minimization Technique [10]. It has recently been modified to include the calculation of piecewise linear parametric bounds on the optimal value function of classes of nonlinear programming problems for which this function is convex or concave. In addition, it can calculate similar bounds on a class of separable non-convex right-hand side programs, once appropriate convex overestimating and underestimating programs are formulated.

SENSUMT is the outgrowth of a routine [3] which first implemented a penalty function sensitivity approach. This was motivated by results given by Fiacco and McCormick [10] for a particular class of perturbations. The algorithmic procedure for this routine was suggested by Fiacco. It was later refined and recoded by Mylander [12] making use of

several subroutines from the SUMT-Version 4 computer program coded by Mylander, Holmes, and McCormick [13]. These routines were subsequently completely integrated with SUMT-Version 4 by Armacost and Mylander [7]. Armacost [1] further revised the program to implement the generalized version of the sensitivity theory developed by Fiacco [9] for general parameter perturbations. This routine, again following a procedure suggested by Fiacco, was recently modified by Ghaemi to conduct piecewise linear parametric bound calculation on the optimal value function of certain classes of nonlinear programming problems subject to large parametric changes [11].

This latter version of the model, designated "SENSUMT," is compiled at the Center for Academic and Administrative Computing of The George Washington University.

The various sections of this manual are arranged as follows. Section 2 reviews the basic sensitivity results and presents the steps of the algorithm implementing sensitivity calculation via a sequential unconstrained minimization algorithm.

Section 3 briefly reviews the procedure for calculating piecewise linear parametric bounds on the optimal value function of certain classes of parametric nonlinear programming problems. It presents the relevant algorithms that have been implemented.

Section 4 gives the details regarding the procedure for coding the problems for solution, sensitivity analysis, and bound calculation with SENSUMT.

Section 5 provides the codes and corresponding computer solution for two illustrative examples that are taken from [11]. For clarity, an annotated computer listing of the input and output for the first example is also provided.

Section 6 gives a description and listing of the various subroutines comprising SENSUMT. This section is included to make the

manual self-contained. With the exception of the subroutines BOUND, PERT, and TRANS, and the new version of program MAIN, the material in this section with a few minor modifications has been taken from [1] and [13].

2. Sensitivity Analysis in Nonlinear Programming

2.1 Basic Sensitivity Results

The parametric mathematical programming problem considered by Fiacco [9] is of the following general form:

$$\begin{aligned} &\underset{x\in E^n}{\text{minimize}} \quad f(x,\varepsilon) \\ &\text{subject to} \quad g_i(x,\varepsilon) \geq 0 \ , \quad i=1,\ldots,m \ , \\ &\qquad\qquad\ \ h_j(x,\varepsilon) = 0 \ , \quad j=1,\ldots,p \ , \end{aligned} \tag{P(ε)}$$

where x is the usual vector of variables and ε is a k-component vector of numbers called "parameters." It is desired to analyze the behavior of a solution vector $x(\varepsilon)$ and the optimal solution value $f^*(\varepsilon) \equiv f[x(\varepsilon),\varepsilon]$ near some given value of ε. Without loss of generality, assume that the parameter vector of interest is $\varepsilon=0$.

The Lagrangian for Problem P(ε) is defined as

$$L(x,u,w,\varepsilon) \equiv f(x,\varepsilon) - \sum_{i=1}^{m} u_i g_i(x,\varepsilon) + \sum_{j=1}^{p} w_j h_j(x,\varepsilon) \ . \tag{2.1}$$

The sensitivity results are based on the following four assumptions:

A1 — The functions defining Problem P(ε) are twice continuously differentiable in (x,ε) in a neighborhood of $(x^*,0)$.

A2 — The second order sufficient conditions for a local minimum of Problem P(0) hold at x^* with associated Lagrange multipliers y^* and w^*.

A3 — The gradients $\nabla_x g_i(x^*,0)$, for all i such that $g_i(x^*,0) = 0$, and $\nabla_x h_j(x^*,0)$, $j=i,\ldots,p$ are linearly independent.

A4 — Strict complementary slackness holds at x^* when $\varepsilon=0$ [i.e., $u_i^* > 0$ for all i such that $g_i(x^*,0) = 0$].

Under the above assumptions, Fiacco [9] established the following generalization of Theorem 6 in [10].

Lemma 2.1 [local characterization of a Kuhn-Tucker triple]: If assumptions A1, A2, A3, and A4 hold for Problem $P(\varepsilon)$ at $(x^*,0)$, then

(a) x^* is a local isolated minimizing point of Problem P(0) and the associated Lagrange multipliers u^* and w^* are unique;

(b) for ε in a neighborhood of 0, there exists a unique once continuously differentiable vector function $y(\varepsilon) = (x(\varepsilon),u(\varepsilon),w(\varepsilon))^T$ satisfying the second order sufficient conditions for a local minimum of Problem $P(\varepsilon)$ such that $y(0) = (x^*,u^*,w^*)^T = y^*$, and hence, $x(\varepsilon)$ is a locally unique, local minimum of Problem $P(\varepsilon)$ with associated unique Lagrance multipliers $u(\varepsilon)$ and $w(\varepsilon)$; and

(c) for ε near 0, the set of binding inequalities is unchanged, strict complementary slackness continues to hold, and the binding constraint gradients are linearly independent at $x(\varepsilon)$.

(d) (Armacost and Fiacco [3]), for ε near 0, the gradient of the optimal value function is

$$\nabla_\varepsilon f^*(\varepsilon) = \nabla_\varepsilon L(y(\varepsilon),\varepsilon) \ , \tag{2.2}$$

(e) which also means that, for ε near 0, the Hessian of the optimal value function is

$$\nabla^2_\varepsilon f^*(\varepsilon) = \nabla^2_\varepsilon L(y(\varepsilon),\varepsilon) \ . \tag{2.3}$$

The above results provide a characterization of a local solution of Problem $P(\varepsilon)$ and its associated optimal Lagrange multipliers near $\varepsilon=0$. They show that the Kuhn-Tucker tripie $y(\varepsilon)$ is unique and well behaved, under the given conditions. Since $y(\varepsilon)$ is once differentiable, the partial derivatives of the components of $y(\varepsilon)$ are well defined. This fact and assumption A1 also mean that the functions defining Problem $P(\varepsilon)$ are once continuously differentiable functions of ε along the "solution trajectory" $x(\varepsilon)$ near $\varepsilon=0$, and the Lagrangian is a once continuously differentiable function of ε along the "Kuhn-Tucker point trajectory." The above results constitute the structure for numerous developments and extensions, many of which have been established by Fiacco [9] and Armacost and Fiacco [2 - 6].

The realization of this theorem for the parametric right hand side problem of special interest in the present study is treated in detail by Armacost and Fiacco [4]. The parametric right hand side problem is the following important realization of $P(\varepsilon)$:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g_i(x) \geq \varepsilon_i \quad , \quad i=1,\ldots,m \ , \\ &&& h_j(x) = \varepsilon_{j+m} \quad , \quad j=1,\ldots,p \ . \end{aligned} \qquad R(\varepsilon)$$

The Lagrangian for $R(\varepsilon)$ is

$$L(x,u,w) \equiv f(x) - \sum_{i=1}^{m} u_i\left(g_i(x)-\varepsilon_i\right) + \sum_{j=1}^{p} w_j\left(h_j(x)-\varepsilon_{j+m}\right) .$$

As is evident from the Lagrangian, the results (d) and (e) of Lemma 2.1 for Problem $R(\varepsilon)$ simplify respectively to

$$(d') \quad \nabla_\varepsilon f^*(\varepsilon) = \begin{bmatrix} u(\varepsilon) \\ -w(\varepsilon) \end{bmatrix}^T \tag{2.4}$$

and

$$(e') \quad \nabla^2_\varepsilon f^*(\varepsilon) = \begin{bmatrix} \nabla_\varepsilon u(\varepsilon) \\ -\nabla_\varepsilon w(\varepsilon) \end{bmatrix} . \tag{2.5}$$

Fiacco [9] has shown that the class of algorithms based on twice continuously differentiable penalty functions (specifically, using the logarithmic-quadratic loss penalty function) can be used to estimate $y(\varepsilon)$ and its derivatives in a neighborhood of $\varepsilon=0$, for the general problem $P(\varepsilon)$. Minimization of the penalty function with penalty parameter r yields a solution of a perturbation of the Kuhn-Tucker system in a neighborhood of $(\varepsilon,r) = (0,0)$. Armacost and Fiacco [3] define an optimal value penalty function and obtain first- and second-order sensitivity estimates which converge to the corresponding sensitivities for the optimal value function for Problem $P(\varepsilon)$.

The logarithmic-quadratic penalty function is

$$W(x,\varepsilon,r) \equiv f(x,\varepsilon) - r \sum_{i=1}^{m} \ln g_i(x,\varepsilon) + (1/2r) \sum_{j=1}^{p} h_j^2(x,\varepsilon) \ . \quad (2.6)$$

Lemma 2.2 (Fiacco [9, Theorem 3.1]): If the assumptions A1 through A4 hold, then in a neighborhood of $(\varepsilon,r) = (0,0)$ there exists a unique once continuously differentiable vector function $y(\varepsilon,r) = [x(\varepsilon,r),u(\varepsilon,r),w(\varepsilon,r)]^T$ satisfying

$$\begin{aligned} \nabla_x L(x,u,w,\varepsilon) &= 0 \quad , \\ u_i g_i(x,\varepsilon) &= r \quad , \quad i=1,\ldots,m \ , \\ h_j(x,\varepsilon) &= w_j r \ , \quad j=1,\ldots,p \ , \end{aligned} \quad (2.7)$$

with $y(0,0) = (x^*,u^*,w^*)$, and such that for any (ε,r) near $(0,0)$ and $r>0$, $x(\varepsilon,r)$ is a locally unique unconstrained local minimizing point of $W(x,\varepsilon,r)$, with $g_i[x(\varepsilon,r),\varepsilon] > 0$, $i=1,\ldots,m$, and $\nabla_x^2 W[x(\varepsilon,r),\varepsilon,r]$ positive definite.

The relevance of Equations (2.7) is the fact that, under the given conditions, when $r=0$, they are necessary conditions that must hold at a local solution of $P(0)$ and, with $r>0$, they are necessary conditions for an unconstrained minimum of $W(x,\varepsilon,r)$. The latter fact can be made obvious by solving for u_i and w_j in (2.7) and obtaining

$$\nabla_x L(x,u,w,\varepsilon) = \nabla_x f - \sum u_i \nabla_x g_i + \sum w_j \nabla_x h_j$$

$$= \nabla_x f - r \sum (1/g_i)\, \nabla_x g_i + (1/r) \sum h_j \nabla_x h_j \qquad (2.8)$$

$$= \nabla_x W(x,\varepsilon,r) \ .$$

Thus, if $y(\varepsilon,r)$ is a solution of (2.7), then

$$\nabla_x W[x(\varepsilon,r),\varepsilon,r] = \nabla_x L[x(\varepsilon,r),u(\varepsilon,r),w(\varepsilon,r),\ ] = 0 \ . \qquad (2.9)$$

This explicit connection between the optimality conditions of local solutions of $P(\varepsilon)$ and unconstrained minima of $W(x,\varepsilon,r)$ makes it possible to approximate information characterizing a local solution of $P(\varepsilon)$ by algorithmic calculations associated with utilizing $W(x,\varepsilon,r)$ to solve $P(\varepsilon)$. In particular, differentiating (2.9) with respect to ε yields

$$\nabla_x^2 W \ \nabla_\varepsilon x + \nabla_{\varepsilon x}^2 W = 0 \ ,$$

and using the fact that $\nabla_x^2 W$ is positive definite (a conclusion of Lemma 2.2) yields

$$\nabla_\varepsilon x = -\nabla_x^2 W^{-1} \ \nabla_{\varepsilon x}^2 W \ , \qquad (2.10)$$

evaluated, of course, at $x(\varepsilon,r)$. Given $\nabla_\varepsilon x(\varepsilon,r)$, the derivatives of the multipliers, $\nabla_\varepsilon u_i(\varepsilon,r)$ and $\nabla_\varepsilon w_j(\varepsilon,r)$, can then be calculated by differentiating the last two systems of equations of (2.7) at $x(\varepsilon,r)$ with respect to ε .

Lemma 2.3 (Fiacco [9]): For ε in a neighborhood of $\varepsilon=0$, it follows that

(a) $\lim_{r\to 0^+} y(\varepsilon,r) = y(\varepsilon,0) = y(\varepsilon)$, the Kuhn-Tucker triple characterized in conclusion (b) of Lemma 2.1; and

(b) $\lim_{r\to 0^+} \nabla_\varepsilon y(\varepsilon,r) = \nabla_\varepsilon y(\varepsilon,0) = \nabla_\varepsilon y(\varepsilon)$.

Under the conditions of Lemma 2.1, $x(\varepsilon,r)$ is a locally unique minimizing point of $W(x,\varepsilon,r)$. Define the "optimal value penalty function" as

$$W^*(\varepsilon,r) \equiv W(x(\varepsilon,r),\varepsilon,r) . \tag{2.11}$$

Armacost and Fiacco [2, Theorem 4 and Corollary 4.1] have obtained further results useful for estimating the first- and second-order sensitivity of the optimal value function $f^*(\varepsilon)$ of Problem $P(\varepsilon)$.

Lemma 2.4 (Armacost and Fiacco [3]): If the assumptions A1 through A4 hold for Problem $P(\varepsilon)$, then in a neighborhood of $\varepsilon=0$,

(a) $\lim_{r\to 0^+} W^*(\varepsilon,r) = f^*(\varepsilon)$;

(b) $\nabla_\varepsilon W^*(\varepsilon,r) = \nabla_\varepsilon L(y,\varepsilon)$ at $y = y(\varepsilon,r)$;

(c) $\lim_{r\to 0^+} \nabla_\varepsilon W^*(\varepsilon,r) = \nabla_\varepsilon f^*(\varepsilon)$;

(d) $\nabla^2_\varepsilon W^*(\varepsilon,r) = \nabla^2_\varepsilon L(y(\varepsilon,r),\varepsilon)$; and

(e) $\lim_{r\to 0^+} \nabla^2_\varepsilon W^*(\varepsilon,r) = \nabla^2_\varepsilon f^*(\varepsilon)$;

where convergence is component by component in all cases.

Lemmas 2.1 through 2.4 enable us to calculate an estimate of $y(\varepsilon)$, $\nabla_\varepsilon y(\varepsilon)$, $\nabla_\varepsilon f^*(\varepsilon)$, and $\nabla^2_\varepsilon f^*(\varepsilon)$ when ε is near 0 and r is near 0, once $y(\varepsilon,r)$ is available.

In the next section we briefly present the algorithmic implementation of some of the above results.

2.2 The Algorithm Implementing the Sensitivity Results

The penalty function algorithm SUMT estimates the solution of the general mathematical problem $P(\varepsilon)$ by estimating the unconstrained minima

of the penalty function $W(x,\varepsilon,r)$ at successively decreasing values of the penalty function parameter $r > 0$. Under conditions weaker than those assumed here, Fiacco and McCormick [10] have shown that as r approaches zero, the sequence of the unconstrained minima of $W(x,\varepsilon,r)$ will approach a solution of $P(\varepsilon)$. Each unconstrained penalty function minimization is thus a "subproblem" associated with a particular value of the penalty function parameter r.

The successive steps of the algorithm for first order sensitivity analysis of the Kuhn-Tucker triple with respect to the j*th* parameter that are implemented in SENSUMT [1] are listed below. Here the notation $\bar{x}$ or $\bar{x}(\bar{\varepsilon})$ denotes the estimate of a solution point of $P(\bar{\varepsilon})$ calculated by SUMT for a given value of the penalty function parameter r, where $\bar{\varepsilon}$ denotes the value of the problem parameter at which this sensitivity is estimated.

Algorithm 2.1:

Step 1. Compute a representation of $\nabla_x^2 W^{-1} = \nabla_x^2 W(\bar{x},\bar{\varepsilon},r)^{-1}$ by L-U decomposition using the SUMT subroutines. If $\nabla_x^2 W$ is not positive definite, terminate the sensitivity analysis.

Step 2. Estimate $\partial(\nabla_x W^T)/\partial\varepsilon_j$ using the central differencing formula

$$\partial(\nabla_x W^T)/\partial\varepsilon_j \approx \left(\frac{1}{2\Delta}\right)\left(\nabla_x W(\bar{x},\bar{\varepsilon}+\Delta e_j,r)^T - \nabla_x W(\bar{x},\bar{\varepsilon}-\Delta e_j,r)^T\right) ,$$

where Δ is the differencing interval and e_j is the j*th* unit vector.

Step 3. Calculate

$$\partial\bar{x}(\bar{\varepsilon})/\partial\varepsilon_j = -\nabla_x^2 W^{-1}\, \partial(\nabla_x W^T)/\partial\varepsilon_j .$$

Step 4. Estimate $\nabla_\varepsilon g_i(\bar{x},\bar{\varepsilon})$ and $\nabla_\varepsilon h_i(\bar{x},\bar{\varepsilon})$ using

$$\partial g_i(\bar{x},\bar{\varepsilon})/\partial\varepsilon_j \approx \left(\frac{1}{2\Delta}\right)\left(g_i(\bar{x},\bar{\varepsilon}+\Delta e_j) - g_i(\bar{x},\bar{\varepsilon}-\Delta e_j)\right) , \text{ and}$$

$$\partial h_i(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j \approx \left(\frac{1}{2\Delta}\right)\left(h_i(\bar{x},\bar{\varepsilon}+\Delta e_j) - h_i(\bar{x},\bar{\varepsilon}-\Delta e_j)\right) .$$

Step 5. Estimate the components of $\nabla_\varepsilon u(\bar{\varepsilon})$ for $i=1,\ldots,m$ using

$$\partial u_i(\bar{\varepsilon})/\partial \varepsilon_j \approx -\left(r/g_i(\bar{x},\bar{\varepsilon})^2\right)\left(\nabla_x g_i(\bar{x},\bar{\varepsilon})\partial \bar{x}(\bar{\varepsilon})/\partial \varepsilon_j + \partial g_i(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j\right) .$$

Step 6. Estimate the components of $\nabla_\varepsilon w(\bar{\varepsilon})$ for $i=1,\ldots,p$ using

$$\partial w_i(\bar{\varepsilon})/\partial \varepsilon_j \approx (1/r)\left(\nabla_x h_i(\bar{x},\bar{\varepsilon})\partial \bar{x}(\bar{\varepsilon})/\partial \varepsilon_j + \partial h_i(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j\right).$$

There are two methods for estimating $\nabla_\varepsilon f^*(\bar{\varepsilon})$, the first using $\nabla_\varepsilon f^* = \nabla_x f \nabla_\varepsilon x + \nabla_\varepsilon f$, with $\nabla_\varepsilon x$ obtained from Step 3, and the second method using the gradient of the penalty function W, or equivalently, the Lagrangian taken with respect to the parameters [Lemma 2.1, conclusion (d); Lemma 2.4, conclusion (b)]. Both are incorporated in the computer program but used for different purposes. The former method gives the most accurate estimate of $\nabla_\varepsilon f^*(\bar{\varepsilon})$ and is summarized in Steps 7 and 8.

Step 7. Estimate the components of $\nabla_\varepsilon f(\bar{x},\bar{\varepsilon})$ using the central differencing formula

$$\partial f(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j \approx \left(\frac{1}{2\Delta}\right)\left(f(\bar{x},\bar{\varepsilon}+\Delta e_j) - f(\bar{x},\bar{\varepsilon}-\Delta e_j)\right) .$$

Calculate an estimate of the components of $\nabla_\varepsilon f^*(\varepsilon)$ using the results of Steps 3 and 7 as

$$\partial f^*(\bar{\varepsilon})/\partial \varepsilon_j \approx \nabla_x f(\bar{x},\bar{\varepsilon})\partial \bar{x}(\bar{\varepsilon})/\partial \varepsilon_j + \partial f(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j .$$

The second method, using the gradient of the Lagrangian to estimate $\nabla_\varepsilon f^*(\bar{\varepsilon})$, is computationally less expensive and is used to obtain rough estimates that single out the more crucial parameters for further analysis. This approximation is calculated as follows.

Step 9. Estimate the components of $\nabla_{\varepsilon} f^*(\bar{\varepsilon})$ using the results of Steps 4 and 7 as

$$\partial f^*(\bar{\varepsilon})/\partial \varepsilon_j \approx \partial f(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j - \sum_{i=1}^{m} u_i(\bar{\varepsilon},r)\partial g_i(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j + \sum_{i=1}^{p} w_i(\bar{\varepsilon},r)\partial h_i(\bar{x},\bar{\varepsilon})/\partial \varepsilon_j .$$

3. Parametric Optimal Value Bounds

In this section we briefly review the procedure for calculating piecewise linear parametric upper and lower bounds on the optimal value function of convex problems with right hand side perturbations of the form $CR(\varepsilon)$. It is important to note that the given technique is not only applicable to problems of this form, but also to any class of problems which have convex or concave optimal value functions. In addition, we show how the above technique can be exterded to calculate similar bounds on the optimal value function of separable nonconvex right hand side perturbation problems $SR(\varepsilon)$, once their (computable) over- and underestimating problems are available. For a more detailed treatment, readers should refer to [11].

3.1 Parametric Bounds on the Optimal Value Function of Convex Right Hand Side Problems $CR(\varepsilon)$

Consider the following right hand side parametric programming problem $R(\varepsilon)$, discussed in Section 2:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \geq \varepsilon_i , \quad i=1,m \\ & h_j(x) = \varepsilon_i , \quad j=m+1,p \end{aligned} \qquad R(\varepsilon)$$

where $x \varepsilon E^n$, f , g , and $h : E^n \to E^1$, and C^2 and $\varepsilon \varepsilon E^p$. If $f(x)$ and $-g_i(x)$, $i=1,m$, are convex and $h_j(x)$, $j=m+1,p$ are linear, then

the problem $R(\varepsilon)$ is convex and will be designated by $CR(\varepsilon)$. It is well known that $f^*(\varepsilon)$, the optimal value function of the problem $CR(\varepsilon)$, is a convex function of ε.

The convexity of the optimal value function $f^*(\varepsilon)$ of the problem $CR(\varepsilon)$ enables one to calculate parametric upper and lower bounds on this function when any of the problem parameters is radically perturbed. This of course requires the solution and corresponding optimal value function sensitivity information for both perturbed and unperturbed problems. Before delving into the calculation procedure of these bounds, we must remind ourselves of the following two basic properties of convex functions.

(i) Any line connecting two points on the graph of a convex function does not underestimate that function between the points.

(ii) Any line tangent to the graph of a convex function does not overestimate that function.

These two properties lend themselves in a natural way to the calculation of parametric bounds on the optimal value function of the problem $CR(\varepsilon)$ under large perturbations of any of the problem parameters, say ε_i.

3.2 Algorithm for Calculation of Bounds on $f^*(\varepsilon)$ of the Problem $CR(\varepsilon)$

In the following we list the successive steps required in calculation of bounds on $f^*(\varepsilon)$, the optimal value function of $CR(\varepsilon)$, as a function of ε_i, when ε_i is perturbed from $\bar{\varepsilon}_{i1}$ to $\bar{\varepsilon}_{i2}$ while the remaining parameters are fixed at their base values.

Algorithm 3.1:

Step 1. Solve the unperturbed problem and obtain $f^*(\varepsilon_i)$ and $df^*(\varepsilon_i)/d\varepsilon_i$ at $\varepsilon_i = \bar{\varepsilon}_{i1}$. Note that

$$\frac{df^*(\varepsilon_i)}{d\varepsilon_i} = \begin{cases} u_i(\varepsilon_i) , & i=1,\ldots,m \qquad (2.4) \\ -w_i(\varepsilon_i) , & i=m+1,p . \qquad (2.5) \end{cases}$$

Step 2. Re-solve the problem and obtain $f^*(\varepsilon_i)$ and $df^*(\varepsilon_i)/d\varepsilon_i$ at $\varepsilon_i = \bar{\varepsilon}_{i2}$.

Step 3. Derive the equation of the line passing through the points $(\varepsilon_i = \bar{\varepsilon}_{i1} , f^*(\varepsilon_i) = f^*(\bar{\varepsilon}_{i1}))$ and $(\varepsilon_i = \bar{\varepsilon}_{i2} , f^*(\varepsilon_i) = f^*(\bar{\varepsilon}_{i2}))$. This line provides a parametric upper bound $\bar{f}^*(\varepsilon_i)$ for $f^*(\varepsilon_i)$ as a function of $\varepsilon_i \in [\bar{\varepsilon}_{i1} , \bar{\varepsilon}_{i2}]$.

Step 4. Derive the equation of the tangent lines to $f^*(\varepsilon_i)$ at the above two points with the slopes

$$\left.\frac{df^*(\varepsilon_i)}{d\varepsilon_i}\right|_{\varepsilon_i=\bar{\varepsilon}_{i1}} \quad \text{and} \quad \left.\frac{df^*(\varepsilon_i)}{d\varepsilon_i}\right|_{\varepsilon_i=\bar{\varepsilon}_{i2}}$$

calculated in Steps 1 and 2, respectively. These two lines provide a parametric lower bound $\underline{f}^*(\varepsilon_i)$ for $f^*(\varepsilon_i)$ as a function of $\varepsilon_i \in [\bar{\varepsilon}_{i1} , \bar{\varepsilon}_{i2}]$.

The lines obtained in Steps 3 and 4 form a triangle which encloses the optimal value function $f^*(\varepsilon_i)$ over the given range of ε_i . An estimate of $f^*(\varepsilon_i)$ can also be made by fitting a convex function that passes through the points given in Step 3 and having the corresponding slopes at these points obtained in Steps 1 and 2.

It is clear that the fundamental requirement in using the above algorithm is that the optimal value function be convex and differentiable at least at $\varepsilon_i = \bar{\varepsilon}_{i1}$ and $\varepsilon_i = \bar{\varepsilon}_{i2}$. As it was given in Section 2, Fiacco, under the assumptions A1 - A4 of Lemma 2.1, has established the differentiability of the optimal value function in a neighborhood of $\varepsilon=0$ for the general parametric programming problem $P(\varepsilon)$. Thus, not

only Problem CR(ε), but all classes of parametric nonlinear programming problems possessing a convex optimal value function and, in particular, meeting assumptions A1 - A4 at $\varepsilon_i = \bar{\varepsilon}_{i1}$ and $\varepsilon_i = \bar{\varepsilon}_{i2}$, can be analyzed for bounds according to Algorithm 3.1.

3.3 Parametric Bounds on the Optimal Value Function of the Problem SR(ε)

In this section we show how Algorithm 3.1 can be extended to calculate piecewise linear parametric upper and lower bounds on the optimal value function of a separable nonconvex right hand side perturbation problem, given the over- and underestimating problems.

3.3.1 Separable right hand side perturbation problems.

The separable nonconvex problem addressed here has the following structure:

$$\begin{aligned} &\underset{x\in E^n}{\text{minimize}} \quad f(x) = \sum_{j=1}^{n} f_j(x_j) \\ &\text{subject to} \quad g_i(x) = \sum_{j=1}^{n} g_{ij}(x_j) \geq \varepsilon_i \,, \quad i=1,\ldots,m \qquad \text{SR}(\varepsilon) \\ &\qquad\qquad L_j \leq x_j \leq U_j \,, \quad j=1,\ldots,n \,, \end{aligned}$$

where each $f_j(x_j)$ and $g_{ij}(x_j)$ is a function of a single variable x_j and is differentiable; L_j and U_j are lower and upper bounds on the variable x_j, respectively. Note that the discussion that follows is also valid when Problem SR(ε) involves linear equality constraints. Let

$$G \equiv \left\{ x : g_i(x) = \sum_{j=1}^{n} g_{ij}(x_j) \geq \varepsilon_i \,, \; i=1,\ldots,m \right\} ,$$

$$B_j \equiv \left\{ x_j : L_j \leq x_j \leq U_j \right\} , \; j=1,\ldots,n \,,$$

$$B \equiv \underset{j=1}{\overset{n}{\times}} B_j \,,$$

and

$$S \equiv G \cap B .$$

The set G, and therefore set S, depends on ε. Thus the problem SR(ε) will read as follows:

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad f(x) = \sum_{j=1}^{n} f_j(x_j) \\ &\text{subject to} \quad x \in B \cap G \equiv S \subset E^n . \end{aligned} \qquad \text{SR}(\varepsilon)$$

3.3.2 Convex underestimating problem.

Let $\underset{\sim}{f}_j(x_j)$ be the convex envelope of $f_j(x_j)$, $j=1,\ldots,m$ and $\tilde{g}_{ij}(x_j)$ be the concave envelope of $g_{ij}(x_j)$, $j=1,\ldots,n$, $i=1,\ldots,m$ over the interval B_j. Thus, for $x_j \in B_j$,

$$\underset{\sim}{f}_j(x_j) \leq f_j(x_j)$$

and

$$\tilde{g}_{ij}(x_j) \geq g_{ij}(x_j) , \quad i=1,\ldots,m; \; j=1,\ldots,n .$$

Definition: Define the problem $\underset{\sim}{\text{CSR}}(\varepsilon)$ as follows:

$$\begin{aligned} &\underset{x \in E^n}{\text{minimize}} \quad \underset{\sim}{f}(x) = \sum_{j=1}^{n} \underset{\sim}{f}_j(x_j) \\ &\text{subject to} \quad x \in G \cap B \equiv \tilde{S} , \end{aligned}$$

where

$$\tilde{G} \equiv \left\{ x : \tilde{g}_i(x) = \sum_{j=1}^{n} \tilde{g}_{ij}(x) \geq \varepsilon_i , \; i=1,\ldots,m \right\} .$$

It follows easily that problem $\underset{\sim}{\text{CSR}}(\varepsilon)$ is a separable convex RHS perturbation problem whose optimal value function underestimates the global optimal value function of the problem SR(ε).

Thus the basic ingredients required to formulate the convex underestimating problem $\underset{\sim}{\text{CSR}}(\varepsilon)$ are the convex envelope of each term in the

objective function and the concave envelope of each term in the constraints of the problem SR(ε) over the rectangular polytope B .

In the next section we will discuss the formulation of a convex overestimating problem C$\tilde{S}$R(ε) of the problem SR(ε).

3.3.3 Convex overestimating problem.

Let $\underset{\approx}{f}_j(x_j)$ be a convex function which does not underestimate $f_j(x_j)$, j=1,...,n , and $\tilde{\tilde{g}}_{ij}(x_j)$ be a concave function which does not overestimate $g_{ij}(x_j)$, j=1,...,n , i=1,...,m , over the interval B_j . Thus, for $x_j \in B_j$,

$$\underset{\approx}{f}_j(x_j) \geq f_j(x_j)$$

and

$$\tilde{\tilde{g}}_{ij}(x_j) \leq g_{ij}(x_j) , \quad i=1,...,m; \; j=1,...,n .$$

Henceforth in discussion, $\underset{\approx}{f}_j(x_j)$ will be abbreviated as convex "nuf" (nonunderestimating function) of the single variable function $f_j(x_j)$, and $\tilde{\tilde{g}}_{ij}(x_j)$ will be abbreviated as concave "nof" (nonoverestimating function) of $g_{ij}(x_j)$ over the interval B_j , i=1,...,m , j=1,...,n .

Definition: Let us define the problem C$\tilde{S}$R(ε) as follows:

$$\underset{x \in E^n}{\text{minimize}} \quad \underset{\approx}{f}(x) = \sum_{j=1}^{n} \underset{\approx}{f}_j(x_j)$$

$$\text{subject to} \quad x \in (\tilde{\tilde{G}} \cap B) \equiv \tilde{\tilde{S}} ,$$

where

$$\tilde{\tilde{G}} \equiv \left\{ x : \tilde{\tilde{g}}_i(x) = \sum_{j=1}^{n} \tilde{\tilde{g}}_{ij}(x_j) \geq \varepsilon_i , \; i=1,...,m \right\} .$$

It is easy to show that problem C$\tilde{S}$R(ε) is a separable convex RHS problem whose optimal value function underestimates the global optimal value function of the problem SR(ε).

Thus, in order to formulate an overestimating convex problem $C\tilde{S}R(\varepsilon)$ of the problem $SR(\varepsilon)$, one must calculate convex nuf, $\underset{\approx}{f}_j(x_j)$, for each component of $f(x)$ and concave nof, $\tilde{\tilde{g}}_{ij}(x_j)$, for each component of $g_i(x)$, $i=1,\ldots,m$, over the interval B_j. It must be noted that convex nuf and concave nof of a given function, unlike its envelopes, are not unique. Thus the overestimating problem $C\tilde{S}R(\varepsilon)$ of problem $SR(\varepsilon)$ is not unique. For further treatment of convex nuf and concave nof of single variable functions, readers may refer to [11].

An important point to be noted here is that formulation of the convex problems $C\tilde{S}R(\varepsilon)$ and $\tilde{CS}R(\varepsilon)$ enable one to calculate parametric upper and lower bounds on the (global) optimal value function of the separable nonconvex programming problem $SR(\varepsilon)$.

3.4 Algorithm for Calculation of Bounds on the (Global) Optimal Value Function of the Problem $SR(\varepsilon)$

In the following an algorithm is presented for calculation of upper and lower bounds on the (global) optimal value function of the problem $SR(\varepsilon)$ as a function of each RHS parameter ε_i, where $\varepsilon_i \in [\bar{\varepsilon}_{i1}, \bar{\varepsilon}_{i2}]$. Implicit in this algorithm is the assumption that the overestimating problem $C\tilde{S}R(\varepsilon)$ has a nonempty interior, which is a basic requirement of almost any nonlinear programming algorithm.

Algorithm 3.2:

Step 1: Calculate $\underset{\approx}{f}_j(x_j)$, a convex nuf for each $f_j(x_j)$, $j=1,\ldots,n$, and $\tilde{\tilde{g}}_{ij}(x_j)$, a concave nof for each $g_{ij}(x_j)$, $i=1,\ldots,m$, $j=1,\ldots,n$, in closed interval B_j, and construct the corresponding overestimating problem $C\tilde{S}R(\varepsilon)$.

Step 2: Solve the problem $C\tilde{S}R(\varepsilon)$ at $\varepsilon_i = \bar{\varepsilon}_{i1}$ and $\varepsilon_i = \bar{\varepsilon}_{i2}$ and obtain $\underset{\approx}{f}^*(\varepsilon_i)$ at these values of ε_i.

Step 3: Derive the equation for the line passing through the points $(\varepsilon_i = \bar{\varepsilon}_{i1}\ ,\ \underset{\approx}{f}^*(\varepsilon_i) = \underset{\approx}{f}^*(\bar{\varepsilon}_{i1}))$ and $(\varepsilon_i = \bar{\varepsilon}_{i2}\ ,\ \underset{\approx}{f}^*(\varepsilon_i) = \underset{\approx}{f}^*(\bar{\varepsilon}_{i2}))$. This line provides a parametric upper bound $\bar{f}^*(\varepsilon_i)$ on the (global) optimal value function $f^*(\varepsilon_i)$ of the problem $SR(\varepsilon)$ as a function of ε_i over the interval $[\bar{\varepsilon}_{i1}, \bar{\varepsilon}_{i2}]$.

Step 4: Calculate $\underset{\sim}{f}_j(x_j)$, the convex envelope for each $f_j(x_j)$, $i=1,\ldots,n$ and $\tilde{g}_{ij}(x_j)$, the concave envelope for each $g_{ij}(x_j)$, $i=1,\ldots,m$, $j=1,\ldots,n$ in closed interval B_j, and construct the corresponding underestimating problem $\underset{\sim}{CSR}(\varepsilon)$.

Step 5: Solve the problem $\underset{\sim}{CSR}(\varepsilon)$ at $\varepsilon_i = \bar{\varepsilon}_{i1}$ and $\varepsilon_i = \bar{\varepsilon}_{i2}$ and obtain $\underset{\sim}{f}^*(\varepsilon_i)$ and $\underset{\sim}{u}_i(\varepsilon_i) = d\underset{\sim}{f}^*(\varepsilon_i)/d\varepsilon_i$ at these values of ε_i.

Step 6: Derive the equation for the tangent lines to $\underset{\sim}{f}^*(\varepsilon_i)$ at $\varepsilon_i = \bar{\varepsilon}_{i1}$ and $\varepsilon_i = \bar{\varepsilon}_{i2}$, with respective slopes of $\underset{\sim}{u}_i(\bar{\varepsilon}_{i1})$ and $\underset{\sim}{u}_i(\bar{\varepsilon}_{i2})$ derived in Step 5. These two lines over the interval $[\bar{\varepsilon}_{i1}, \bar{\varepsilon}_{i2}]$ jointly provide a parametric lower bound $\underline{f}^*(\varepsilon_i)$ on the (global) optimal value function $f^*(\varepsilon_i)$ of the problem $SR(\varepsilon)$ as a function of ε_i.

The implicit assumption in Steps 5 and 6 of this algorithm is that assumptions A1 - A4 of Lemma 2.1 hold at the solution points when ε_i takes values of $\bar{\varepsilon}_{i1}$ and $\bar{\varepsilon}_{i2}$.

Before delving into the next section, we must point out that if over- and underestimating problems with concave optimal value functions could be formulated, then the algorithm above, with minor alterations, can be used to calculate the desired bounds. In this instance Step 6, when applied to the overestimating problem, will provide an upper bound

while Step 3, when applied to the underestimating problem, will provide a lower bound on the optimal value function. For calculation of envelopes, convex nuf and concave nof of single variable functions, readers may refer to [11]. The entries in Table 1 summarize the concept of convex nuf and concave nof for a variety of single variable functions.

4. SENSUMT Input Specifications

The coding procedure for nonlinear programming problems for solution, sensitivity analysis, and bound calculation using SENSUMT closely follows the coding procedure for SUMT-Version 4 [13]. In order to code a given problem for the above calculations, one must have (i) a user-supplied subroutine, and (ii) user's information cards.

4.1 User-supplied Subroutines

User-supplied subroutines will generally consist of four subroutines called READIN, RESTNT, GRAD1 and MATRIX. These subroutines are the only subroutines for SENSUMT that are problem dependent. The communication between these subroutines and the rest of the subroutines of SENSUMT is mostly through the COMMON blocks, but partly via their arguments. Each of the user-supplied subroutines must contain the double precision card

```
IMPLICIT REAL*8(A-H,O-Z).
```

The subroutines RESTNT, GRAD1, and MATRIX must contain the COMMON card

```
COMMON/SHARE/X(45), DEL(45), A(45,45), N, M, MN, NP1, NM1.
```

Also, depending on the problem being solved, the following COMMON card may be required in some or all of the user subroutines:

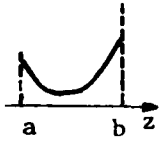
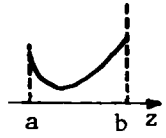
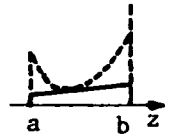
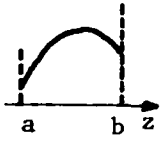
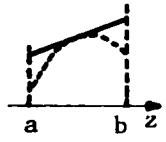
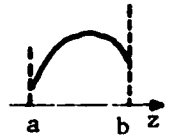
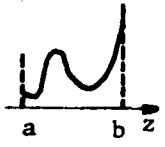
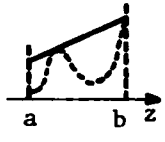
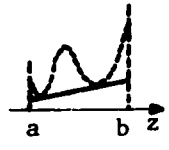
```
COMMON/SEN/PAR(45), DPAR(45), NPAR, ISENS.
```

When bound analysis is desired, subroutine READIN must contain the COMMON card

```
COMMON/ABG/LY, LZ, PER(45).
```

TABLE 1

CONVEX nuf AND CONCAVE nof OF THE SINGLE VARIABLE FUNCTION T(z) OVER THE CLOSED INTERVAL $a \leq z \leq b$

T(z)		$\underset{\approx}{T}(z)$ [convex nuf of T(z)]		$\tilde{\tilde{T}}(z)$ [concave nof of T(z)]	
$a \leq z \leq b$	Graph	$a \leq z \leq b$	Graph	$a \leq z \leq b$	Graph
Convex		$\underset{\approx}{T}(z) = T(z)$		$\tilde{\tilde{T}}(z)$ = any line not overestimating T(z)	
Concave		$\underset{\approx}{T}(z)$ = any line not underestimating T(z)		$\tilde{\tilde{T}}(z) = T(z)$	
Neither convex nor concave		$\underset{\approx}{T}(z)$ = any line not underestimating T(z)		$\tilde{\tilde{T}}(z)$ = any line not overestimating T(z)	

In the following section we discuss the purpose and coding procedure of each of these user-supplied subroutines. The general description and listing of the rest of the subroutines comprising SENSUMT will be given in Section 6.

4.1.1 READIN

This subroutine allows the user to read in the data which is necessary for evaluation of the problem functions and convey it, through the COMMON blocks, to the related subroutines. The problem parameters for which sensitivity and bound calculation are desired must also be defined and initialized in this subroutine. The initial values of the parameters must be placed in array PAR. The corresponding perturbation values for bound calculation must be placed in array PER. NPAR in this subroutine must be initialized to the total number of problem parameters on which sensitivity is desired. *Notice that the presence of array PER in this subroutine will trigger SENSUMT to calculate the desired bound calculation.* READIN may also be used to read in and print out any pertinent information of the user's choice, such as problem title, date, and so on. This subroutine is called only once for each problem being solved. For clarity, see Example 1 and its code in Figure 2. Any input data read by subroutine READIN must be placed immediately after the first option card.

4.1.2 RESTNT (IN,VAL)

This subroutine is used to read in the problem functions. For IN=0, set VAL=f(x) and for IN≠0 set

$$\text{VAL} = g_{\text{IN}}(x) \ , \quad \text{IN}=1,2,\ldots,m$$

$$\text{VAL} = h_{\text{IN}}(x) \ , \quad \text{IN}=m+1,\ldots,m+p \ .$$

For clarity see Example 1 and Figure 2.

4.1.3 GRAD1 (IN)

This subroutine is used to read in the gradients of the problem functions. For IN=0 set

$$DEL(J) = f(x)/\partial x_J \ , \quad J=1,2,\ldots,n$$

and for IN≠0 set

$$DEL(J) = \begin{cases} \partial g_{IN}(x)/\partial x_J \ , & J=1,2,\ldots,n \\ \partial h_{IN}(x)/\partial x_J \ , & J=1,2,\ldots,n \end{cases} .$$

For an illustration, see Example 1 and Figure 2.

SENSUMT can internally compute numerical approximations for some or all of the gradients. To utilize this option, instead of coding the gradients code the statements

```
CALL DIFF1(IN)
RETURN
```

for any desired value of the variable IN (the index of the problem functions, with IN=0 corresponding to the objective function f(x)). Because of the computational effort involved in numerical differencing, this option may be prohibitive for large problems.

4.1.4 MATRIX (IN,K)

This subroutine reads in the *upper triangle and diagonal elements* of the Hessian matrix of the problem functions. For IN=0, set

$$A(I,J) = \partial^2 f(x)/\partial x_I \partial x_J \ , \quad I=1,2,\ldots,n; \ J=1,2,\ldots,n; \text{ and } I \leq J \ .$$

For IN≠0 set

$$A(I,J) = \begin{cases} \partial^2 g_{IN}(x)/\partial x_I \partial x_J \ , & IN=1,2,\ldots,m \\ \partial^2 h_{IN}(x)/\partial x_I \partial x_J \ , & IN=m+1,\ldots,p \end{cases} ,$$

where I=1,2,...,n; J=1,2,...,n; and $I \leq J$. Before the call is made to this subroutine, all entries of the matrix A(I,J) are set to zero.

The second argument K of the subroutine MATRIX is provided so that the user may communicate to SENSUMT that the Hessian of a problem function is identically zero. Set K=1 if the second partial derivatives of the IN*th* constraint are zero (IN=0 corresponds to the objective function). For an illustration, see Example 1 and Figure 2.

SENSUMT can internally compute numerical approximations for some or all of the Hessian matrices. To utilize this option, instead of coding the Hessian matrices, code the statements

```
CALL DIFF2(IN)
RETURN
```

for any desired value of the variable IN (the index of the problem function, with IN=0 designating the objective function). *Although this option can be used in calculating an optimal solution, the current sensitivity routines require explicit coding of the closed form of the Hessian matrices. Thus, this option cannot be used for sensitivity and bound calculations.* Again, because of the computational effort involved in numerical differencing, use of this option is not advised for large problems.

4.2 User's Information Cards

The other inputs required by SENSUMT are the user's information cards, i.e.,

- PARAMETER CARD
- INITIAL VECTOR CARD(S)
- FIRST OPTION CARD
- TOLERANCE CARD
- SECOND OPTION CARD.

In the following we specify the type of information, along with the corresponding format, name, and description, which must be provided by the user to SENSUMT via the above cards. All of this information is read by the program MAIN, which sets up the SENSUMT algorithm to solve and

analyze the problem under study. Tables 2, 3, and 4, which follow, are taken from [13].

4.2.1 Parameter card.

Input specifications of the parameter card are shown in Table 2.

TABLE 2

PARAMETER CARD

Columns	Format	Name	Use
01-12	E12.0	EPSI (ε)	Tolerance used to decide if an unconstrained minimum has been achieved for each subproblem [see Option 9]
13-24	E12.0	RHØIN (r_1)	Possible initial value of r (often set at 1.0) [see Option 1]
25-36	E12.0	THETAO (θ_0)	Tolerance used to decide if the solution to the NLP problem $P(\varepsilon)$ has been approximated [see Option 5]
37-48	E12.0	RATIØ (c)	Parameter (>1) used to compute consecutive values of r; $r_{i+1} = r_i/c$ (often set at 16.0)
49-60	E12.0	TMMAX	Maximum amount of time for solving problem (in seconds)
61-64	I4	M	Number (integer) of nontrivial constraints [see Option 2] $(M+MZ) \leq 200$*
65-68	I4	N	Number (integer) of variables, $N \leq 45$
69-72	I4	MZ	Number (integer) of equality constraints

*The limits on M+MZ and N are governed by the size of the arrays in SENSUMT.

4.2.2 Initial vector card(s).

The cards designating the initial starting point immediately follow the parameter card. There are six components per card, requiring n/6 cards for the initial vector. Each card has the format 6E12.6 .

4.2.3 First option card.

The input specifications for the first option card are shown in Table 3. This card must immediately follow the user-supplied subroutines.

4.2.4 Tolerance card.

The input specifications for the tolerance card are given in Table 4. This card must immediately follow the first option card.

4.2.5 Second option card.

The input specifications for the second option card, which immediately follows the tolerance card, are given in Table 5.

Figure 1 shows the arrangement of the data together with JOB and JCL cards in a coded deck.

TABLE 3

FIRST OPTION CARD

Option	Column	Value	Meaning
1 (normally set to 3)	7	=1	The value for r is made by finding an approximate solution $\min\{[\nabla W(x^0,r)[\nabla^2 W(x^0,r)]^{-1}\nabla W(x^0,r)]$ which is a good approximation only when x^0 is close to the boundary (i.e., for some i, $g_i(x)$ is close to zero) or when $\nabla^2 f(x^0)=0$ and when MZ=0.
		=2	r_1 is given by formula 8.65 [10, p. 191] (can only be used when MZ=0).
		=3	r_1 = RHØIN (see parameter card).
2	14	=1	The requirements (trivial constraints) that $x_i \geq 0$ for i=1,...,n are to be automatically included in the problem.
		=2	The only constraints on the problem are those inputted by the user.
3 (normally set to 1)	21	=1	Standard printout (this includes a call to OUTPUT after the solution of every subproblem). Also the estimates of the "Lagrange multipliers" and first and second order solution estimates are printed.
		=2	For additional printout (includes standard printout and every intermediate point, gradient of P and the vector S).
4 (normally set to 1)	28	=1	Final convergence is determined on the basis of current solution to the subproblem.
		=2	Final convergence is determined on the basis of the first order estimates. The first order estimate of the solution vector must be close to feasible. See below.
		=3	Final convergence is determined on the basis of the second order estimates. The second order estimate of the solution vector must be close to feasible before the convergence check is made. If $\bar{x}$ is a solution

Option	Column	Value	Meaning
			estimate it is considered close to being feasible if $g_i(\bar{x}) + \theta_0 \geq 0$, i=1,2,...,m, where θ_0 is defined on the parameter card.
5 (normally set to 1) (see [10, p. 195])	35		The convergence criterion determining the NLP problem has been solved (only use =1, when NT4≠1).
		=1	Quit when $\frac{G - f[x(r_k)]}{G[x(r_k),\mu(r_k),\lambda(r_k)]} < \theta_0$
		=2	Quit when $\left\| r \sum_{j=1}^{m} \ell ng_j[x(r_k)] \right\| < \theta_0$
		=3	Quit when $\frac{\text{first order estimate of } v_0}{G[x(r_k),\mu(r_k),\lambda(r_k)]} - 1 < \theta_0$
6 (normally set to 1)	42	=1	After final convergence the program reads in new data and solves the next problem.
		=2	After final convergence has been determined a call to PUNCH is made before proceeding on to the next problem.
7 (normally set to 1)	49		First move after a minimum to a subproblem is achieved
		=1	No extrapolation.
		=2	Extrapolate through last two minima.
		=3	Extrapolate through last three minima.
8	56		Not used.
9	63		Subproblem convergence criterion, or when to stop minimizing P function for fixed value of r (see parameter card).
		=1	Quit when $\left\| \nabla_x W^T(x^i,r) \left[\frac{\partial^2 W(x,r)}{\partial x_i \partial x_j}\right]^{-1} \nabla_x W(x^i,r) \right\| < \epsilon.$
		=2	Quit when $\left\| \nabla_x W^T(x^i,r) \left[\frac{\partial^2 W(x,r)}{\partial x_j \partial x_j}\right]^{-1} \nabla_x W(x^i,r) \right\| < \frac{W(x^{i-1}) - W(x^i)}{5}.$

Table 3--*continued*

Option	Column	Value	Meaning
		=3	Quit when $\lvert\nabla_x W(x^i,r)\rvert < \varepsilon$
10[a]	70	=1	At least one nonlinear constraint
		=2	Linear constraints
		=3	Linear constraints and linear objective function (i.e., a linear programming problem)

[a]When option 10=3, MATRIX (the user subroutine supplying the second partial derivatives) will not be called, and when option 10=2 it will be called only to get the second partials of f(x).

TABLE 4

TOLERANCE CARD

Columns	Format	Name	Description
01-12	E12.6	XEP1	If some first or second derivatives are to be gotten by numerical differencing (see addition option card) this is the value used to compute them. See the description of DIFF1. (Usually setting XEP1 equal to .0001 is satisfactory, although a good value is dependent on the scaling of the problem.)
13-24	E12.6	XEP2	When minimizing the W function for a given value of r (RHØ) the value of W must decrease by an amount exceeding XEP2 for each iteration after the first. If it does not, then the code prints out the message "apparently roundoff errors prevent a more accurate determination of the minimum of this subproblem," and it is assumed a minimum has been found. (Usually we set XEP2 equal to 0.)

TABLE 5

SECOND OPTION CARD

Option	Column	Value	Meaning
1 (normally set to 4)	7	=1	Solve problems without checking derivatives
		=2	Solve problem after checking only first derivatives
		=3	Do not solve problem after checking only first derivatives
		=4	Solve problem after checking first and second derivatives
		=5	Do not solve problem after checking first and second derivatives
2	14	=1	The method for minimizing the unconstrained penalty function is to be the generalized Newton-Raphson method as modified to handle indefinite Hessian matrices. This method requires function values, first and second derivatives.
		=2	Same as 1, except that when an "orthogonal move" is made because of an indefinite Hessian matrix, $-\nabla P$ is added to the orthogonal move vector.
		=3	Steepest descent is used to minimize P-function.
		=4	The method for minimizing the unconstrained penalty function is McCormick's modification of the Fletcher-Powell method as reported in [10]. This requires function values and first derivatives.
3	21	=0	Do not conduct a sensitivity analysis.
		=1	Conduct a sensitivity analysis at the final subproblem with a fixed value for the differencing interval.
		=2	Conduct a sensitivity analysis at each subproblem along the minimizing trajectory with the value of the differencing interval depending on the particular subproblem.
		=3	Conduct a sensitivity analysis at the final subproblem for a range of differencing intervals.

Table 5--*continued*

Option	Column	Value	Meaning
4	28	=0	Do not estimate the partial derivatives of the estimates of the Lagrange multipliers.
		=1	Estimate the partial derivatives of the estimates of the Lagrange multipliers whenever a sensitivity analysis of the solution point is conducted.
5	35	=0	Estimate the partial derivatives of the optimal value function and eliminate those parameters which do not affect the optimal value functions from subsequent sensitivity calculations.
		=1	Estimate the partial derivatives of the optimal value function with respect to all parameters, but continue all subsequent sensitivity calculations with respect to all parameters.
		=2	Do not estimate the partial derivatives of the optimal value function first. Conduct the sensitivity analysis with respect to all parameters.
6	42	=0	Do not transform the results.
		=1	The problem being solved, P(x), is a convex equivalent of a geometric programming problem G(t) obtained by transformation $t_i = e^{-xi}$. Thus back transform the results to t space.
7	49	=1	$f^*(\varepsilon)$ is convex. Derive parametric upper and lower bounds on $f^*(\varepsilon)$.
		=2	$f^*(\varepsilon)$ is convex. Derive parametric upper bound on $f^*(\varepsilon)$.
		=3	$f^*(\varepsilon)$ is convex. Derive parametric lower bound on $f^*(\varepsilon)$.
		=4	$f^*(\varepsilon)$ is concave. Derive parametric upper and lower bounds on $f^*(\varepsilon)$.
		=5	$f^*(\varepsilon)$ is concave. Derive parametric upper bound on $f^*(\varepsilon)$.
		=6	$f^*(\varepsilon)$ is concave. Derive parametric lower bound on $f^*(\varepsilon)$.

NOTE: This data deck arrangement assumes that the user subroutines are linked with the rest of the SENSUMT subroutines (which are loaded in a data set) via the JCL cards.

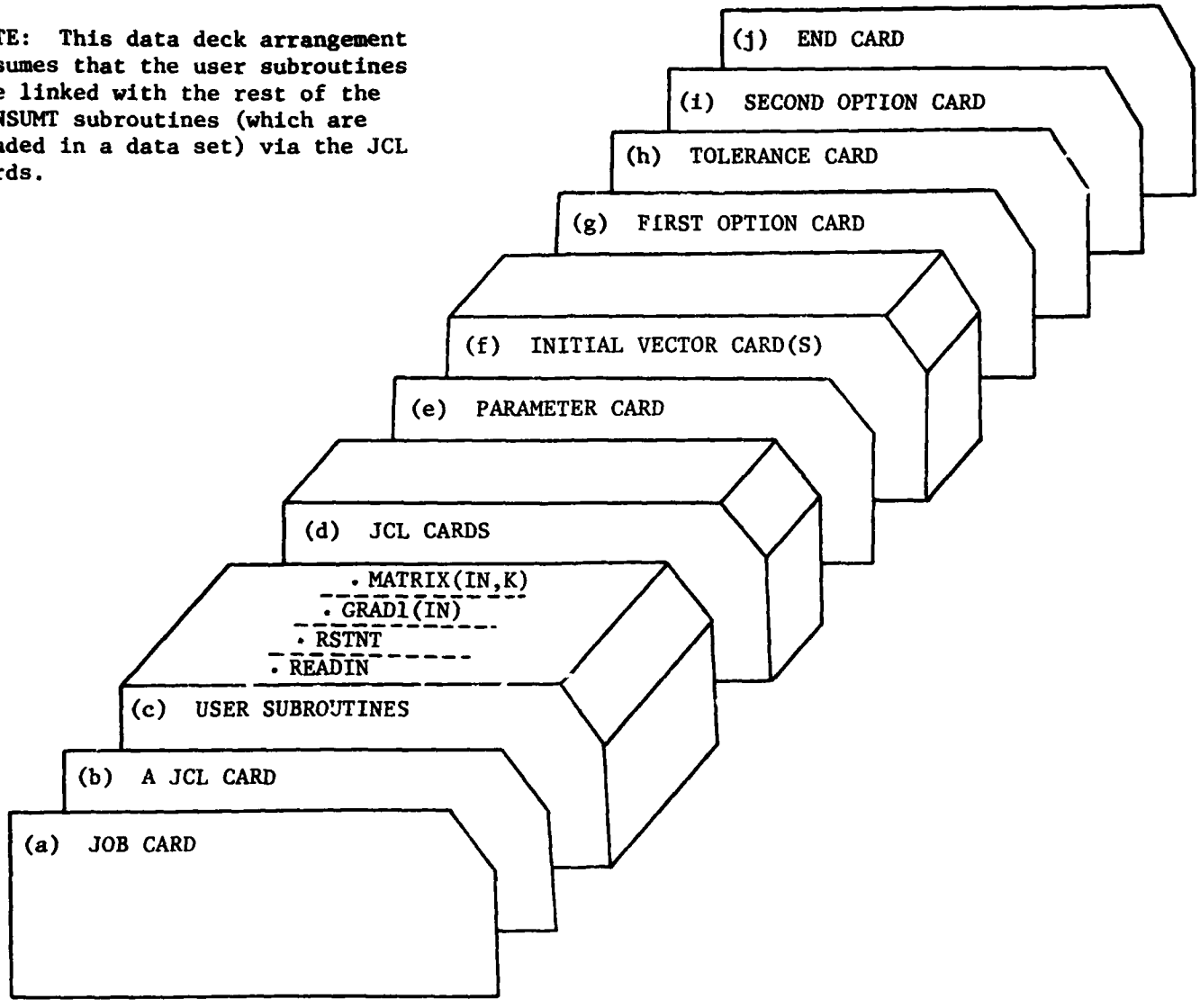


Figure 1.--Data deck structure for SENSUMT.

5. Coded Examples and Input/Output Illustrations

In this section we provide the input listing and output description of two illustrative examples which are taken from [11].

5.1 Example 1

Consider the following convex RHS programming problem:

$$\text{minimize} \quad f(x) = (x_1-4)^2 + (x_2-2)^2$$

$$\text{subject to} \quad g_1(x) = -x_1^2 + x_2 \geq \varepsilon_1$$

$$g_2(x) = -x_1 - x_2 \geq \varepsilon_2 \ .$$

It is desired to solve and analyze this problem for sensitivity when $\varepsilon_1 = 0$ and $\varepsilon_2 = -3$. Moreover, it is desired to derive parametric upper and lower bounds on the optimal value function of this problem when

(i) $-1 \leq \varepsilon_1 \leq 0$ while $\varepsilon_2 = -3$, and

(ii) $-3 \leq \varepsilon_2 \leq -1$ while $\varepsilon_1 = 0$.

5.1.1 Computer listing of the code deck.

Figure 2 shows the computer listing of the code of Example 1. The letters in circles categorizing the input data correspond to those indicated in the code deck structure depicted in Figure 1. The format of the listed data is given in Section 4.

5.1.2 Selected pages from the computer output.

Figure 3 lists an annotated computer output for Example 1. An explanation of the meaning of the corresponding steps follows.

```
//GHAEMI T=3 L=3                                                    (a) Job card
// EXEC FORG1                                                       (b) A JCL card
      SUBROUTINE READIN                                             (c) User subroutines
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      COMMON/ABG/LY,LZ,PER(45)
      NPAR=2
      PAR(1)=0.
      PAR(2)=-3.0
      PER(1)=-1.
      PER(2)=2.
      RETURN
      END                                                           Subroutine READIN
      SUBROUTINE RESINI(IN,VAL)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SHARE/X(45),DEL(45), A(45,45),N,M,MN,NP1,NM1
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      COMMON/ABG/LY,LZ,PER(45)
      I=IN+1
      GO TO (20,1,2),I
   20 VAL=(X(1)-4.)**2+(X(2)-2.)**2
      RETURN
    1 VAL=-X(1)**2+X(2)-PAR(1)
      RETURN
    2 VAL=-X(1)-X(2)-PAR(2)
      RETURN
      END                                                           Subroutine RSTNT (IN,VAL)
      SUBROUTINE GRADI(IN)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SHARE/X(45),DEL(45), A(45,45),N,M,MN,NP1,NM1
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      J=IN+1
      DO 5 I=1,N
      DEL(I)=0.0
    5 CONTINUE
      GO TO (20,1,2),J
   20 DEL(1)=2.*(X(1)-4.)
      DEL(2)=2.*(X(2)-2.)
      RETURN
    1 DEL(1)=-2.*X(1)
      DEL(2)=1.0
      RETURN
    2 DEL(1)=-1.0
      DEL(2)=-1.0
      RETURN
      END                                                           Subroutine GRAD (IN)
      SUBROUTINE MATRIX(IN,K)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SHARE/X(45),DEL(45), A(45,45),N,M,MN,NP1,NM1
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      L=IN+1
      GO TO (20,1,2),L
   20 A(1,1)=2.0
      A(1,2)=0.
      A(2,2)=2.0
      RETURN
    1 A(1,1)=-2.0
      A(1,2)=0.
      A(2,2)=0.
      RETURN
    2 K=1
      RETURN
      END                                                           Subroutine MATRIX (IN,K)
// EXEC FORG6,DSN='OR79661.BOUNDS45',PROG=MAIN                      (d) JCL cards
//GO.SYSLIB  DD
//           DD
//   DD  DSN=GNU.FORTLIB,DISP=SHR
//GO.SYSLIN  DD
//   DD  DSN=&TEMPUNCH,DISP=(OLD,DELETE)
//FT07F001   DD  SYSOUT=A
    1.0E-08          10.0      1.0E-06          4.0        900.0  2  2  0   (e) Parameter card
        1.                1.5                                                (f) Initial vector card
     3     2                                                                 (g) First option card
 0.1E-10     0.00                                                           (h) Tolerance card
                                  0                                         (i) Second option card
//                                                                          (j) End card
```

Figure 2.--Computer listing of the code for Example 1.

```
(I)
NONLINEAR PROGRAMMING ROUTINE-SUMT VERSION 4 03/01/73
(1)   M= 2      M= 2      MZ= 0

      MAX. TIME= 0.90000000E 03    R= 0.10000000E 02    RATIO= 0.40000000E 01      EPSILON= 0.9999997E-08    THETA= 0.9999996E-06
(2) OPTIONS SELECTED
      3      2      1      1      1      1      1      0      1      1
(3) TOLERANCES
      0.0                    0.9999999E-03
(4) SECOND SET OF OPTIONS
      1      1      1      1      1      0      1
(5) TIME=    0.059 SECONDS
(6)   F= 0.92500000 01     P= 0.0                G= 0.0              RSIGMA= 0.0              H= 0.0
      THE CURRENT VALUE OF X IS
       0.10000000 01       0.15000000 01
      THE CONSTRAINT VALUES
       0.50000000 00       0.50000000 00
      TIME=    0.102 SECONDS

(7) *****THE FEASIBLE STARTING POINT TO BE USED IS ...
      F= 0.92500000 01     P= 0.0                G= 0.0              RSIGMA= 0.0              H= 0.0
      THE CURRENT VALUE OF X IS
       0.10000000 01       0.15000000 01
      THE CONSTRAINT VALUES
       0.50000000 00       0.50000000 00
    APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
      TIME=    0.219 SECONDS

(8) **********************************************
      POINT=  4        DDTT= 0.40671930-06      RHO= 0.10000000 02      MAGNITUDE= 0.19341530-02      PHASE= 2
      F= 0.15780290 02     P= 0.80717330 01      G= -0.42197120 01     RSIGMA= -0.77085550 01      H= 0.0
      THE CURRENT VALUE OF X IS
       0.51007220-01       0.15690190 01
      THE CONSTRAINT VALUES
       0.15666170 01       0.13799740 01

(9) LAGRANGE MULTIPLIERS
      F= 0.15780290 02     P= 0.80717330 01      G= -0.42197120 01     RSIGMA= -0.77085550 01      H= 0.0
      THE CURRENT VALUE OF X IS
       0.51007220-01       0.15690190 01
      THE CONSTRAINT VALUES
       0.83839740 01       0.72465150 01
    APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
      TIME=    0.340 SECONDS

(10) **********************************************
      POINT=  7        DDTT= 0.23087850-04      RHO= 0.25000000 01      MAGNITUDE= 0.1893447D-01      PHASE= 2
      F= 0.11111590 02     P= 0.11665690 02      G= 0.61115940 01     RSIGMA= 0.55409590 00      H= 0.0
      THE CURRENT VALUE OF X IS
       0.69826920 00       0.15615600 01
      THE CONSTRAINT VALUES
       0.10539800 01       0.76017110 00

(11) 1ST ORDER ESTIMATES
      F= 0.94748970 01     P= 0.12863680 02      G= 0.95553620 01     RSIGMA= 0.0              H= 0.0
      THE CURRENT VALUE OF X IS
       0.91402110 00       0.15324070 01
      THE CONSTRAINT VALUES
       0.48598830 00       0.55367020 00
```

Figure 3.--Annotated computer output for Example 1.

```
 LAGRANGE MULTIPLIERS
     F=  0.9741894D 01      P=  0.12863680 02      G=  0.9555362D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.9140231D 00          0.1532407D 01
   THE CONSTRAINT VALUES
    0.2371962D 01          0.3288733D 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.469 SECONDS

(12) ********************************************
       POINT=  10       DOTT=  0.7459573D-04      RHO=  0.62500000 00      MAGNITUDE=  0.4996642D-01      PHASE= 2
     F=  0.8511950D 01      P=  0.9906426D 01      G=  0.7261950D 01      RSIGMA=  0.1394476D 01      H=  0.0
   THE CURRENT VALUE OF X IS
    0.1106836D 01          0.1623767D 01
   THE CONSTRAINT VALUES
    0.3986811D 00          0.2693975D 00

(13)  2ND  ORDER ESTIMATES
     F=  0.7596676D 01      P=  0.9083760D 01      G=  0.7645402D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1264958D 01          0.1659086D 01
   THE CONSTRAINT VALUES
    0.5896728D-01          0.7595534D-01

    1ST  ORDER ESTIMATES
     F=  0.7722596D 01      P=  0.9320005D 01      G=  0.7645402D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1243025D 01          0.1651169D 01
   THE CONSTRAINT VALUES
    0.1060585D 00          0.1058063D 00

  LAGRANGE MULTIPLIERS
     F=  0.7722596D 01      P=  0.9320005D 01      G=  0.7645402D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1243025D 01          0.1651169D 01
   THE CONSTRAINT VALUES
    0.1567669D 01          0.2319992D 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.750 SECONDS

(14) ********************************************
       POINT=  13       DOTT=  0.1185880D-03      RHO=  0.15625000 00      MAGNITUDE=  0.1157642D 00      PHASE= 2
     F=  0.7671997D 01      P=  0.8414190D 01      G=  0.7359497D 01      RSIGMA=  0.7421929D 00      H=  0.0
   THE CURRENT VALUE OF X IS
    0.1249429D 01          0.1673876D 01
   THE CONSTRAINT VALUES
    0.1128031D 00          0.7669464D-01

    2ND  ORDER ESTIMATES
     F=  0.7381120D 01      P=  0.7823229D 01      G=  0.7392012D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1300556D 01          0.1693207D 01
   THE CONSTRAINT VALUES
    0.1761123D-02          0.6237300D-02

    1ST  ORDER ESTIMATES
     F=  0.7402165D 01      P=  0.7916778D 01      G=  0.7392012D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1294963D 01          0.1680579D 01
```

Figure 3.--*continued*

```
 CURRENT VALUES
 0.3473e431-02        0.1246036D-01

 LAGRANGE MULTIPLIERS
    F=  0.74021650 01     P=  0.79167780 01     G=  0.73920120 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.12969630 01        0.16905790 01
  THE CONSTRAINT VALUES
   0.13851570 01        0.20373000 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.879 SECONDS

(15) ***********************************************
       POINT=  15       DOTT=  0.75072110-03      RHO=  0.39062500-01      MAGNITUDE=  0.34600050 00      PHASE= 2
    F=  0.74436570 01     P=  0.77354740 01     G=  0.73655320 01     RSIGMA=  0.29181680 00     H=  0.0
  THE CURRENT VALUE OF X IS
   0.12892490 01        0.16909900 01
  THE CONSTRAINT VALUES
   0.28827070-01        0.19760510-01

  2ND  ORDER ESTIMATES
    F=  0.73661310 01     P=  0.74820660 01     G=  0.73675440 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.13028930 01        0.16971030 01
  THE CONSTRAINT VALUES
  -0.42809230-03        0.39366340-05

  1ST  ORDER ESTIMATES
    F=  0.73683790 01     P=  0.75092360 01     G=  0.73675440 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.13025220 01        0.16966950 01
  THE CONSTRAINT VALUES
   0.13032880-03        0.78246310-03

 LAGRANGE MULTIPLIERS
    F=  0.73683790 01     P=  0.75092360 01     G=  0.73675440 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.13025220 01        0.16966950 01
  THE CONSTRAINT VALUES
   0.13550630 01        0.19767970 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    1.000 SECONDS

(16) ***********************************************
       POINT=  17       DOTT=  0.18547470-04      RHO=  0.97656250-02      MAGNITUDE=  0.10391300 00      PHASE= 2
    F=  0.73861610 01     P=  0.74858740 01     G=  0.73666300 01     RSIGMA=  0.99712330-01     H=  0.0
  THE CURRENT VALUE OF X IS
   0.12993460 01        0.16956500 01
  THE CONSTRAINT VALUES
   0.73495540-02        0.50044530-02

  2ND  ORDER ESTIMATES
    F=  0.73669620 01     P=  0.73955690 01     G=  0.73669960 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.13027240 01        0.16972360 01
  THE CONSTRAINT VALUES
   0.14605230-03        0.39322420-04

  1ST  ORDER ESTIMATES
    F=  0.73670510 01     P=  0.74034730 01     G=  0.73669960 01     RSIGMA=  0.0              H=  0.0
```

Figure 3.--*continued*

```
THE CURRENT VALUE OF X IS
 0.13027120 01        0.16972030 01
THE CONSTRAINT VALUES
 0.14507190-03        0.85768720-04


LAGRANGE MULTIPLIERS
     F= 0.73670510 01     P= 0.74026730 01     G= 0.7366996D 01     RSIGMA= 0.0              H= 0.0
   THE CURRENT VALUE OF X IS
    0.13027120 01        0.16972030 01
   THE CONSTRAINT VALUES
    0.13287370 01        0.19513870 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
     TIME=    1.102 SECONDS

(17) ********************************************
        POINT= 19        DOTT= 0.20282430-05      RHO= 0.24414060-02      MAGNITUDE= 0.7027628D-01      PHASE= 2
     F= 0.73715670 01     P= 0.74032510 01     G= 0.7366685D 01     RSIGMA= 0.3168376D-01     H= 0.0
   THE CURRENT VALUE OF X IS
    0.13019160 01        0.16968320 01
   THE CONSTRAINT VALUES
    0.18463760-02        0.1251843D-02


  2ND  ORDER ESTIMATES
     F= 0.73666830 01     P= 0.73739130 01     G= 0.7366703D 01     RSIGMA= 0.0              H= 0.0
   THE CURRENT VALUE OF X IS
    0.13027770 01        0.16972280 01
   THE CONSTRAINT VALUES
   -0.21372170-07       -0.46807200-05


  1ST  ORDER ESTIMATES
     F= 0.7366706D 01     P= 0.73757100 01     G= 0.7366703D 01     RSIGMA= 0.0              H= 0.0
   THE CURRENT VALUE OF X IS
    0.13027730 01        0.16972260 01
   THE CONSTRAINT VALUES
    0.90472090-05        0.97236950-06


LAGRANGE MULTIPLIERS
     F= 0.7366706D 01     P= 0.73757100 01     G= 0.7366703D 01     RSIGMA= 0.0              H= 0.0
   THE CURRENT VALUE OF X IS
    0.13027730 01        0.16972260 01
   THE CONSTRAINT VALUES
    0.13222690 01        0.19502500 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
     TIME=    1.219 SECONDS

(18) ********************************************
        POINT= 21        DOTT= 0.55388060-06      RHO= 0.61035160-03      MAGNITUDE= 0.8034309D-01      PHASE= 2
     F= 0.73679130 01     P= 0.73775240 01     G= 0.7366692D 01     RSIGMA= 0.9611244D-02     H= 0.0
   THE CURRENT VALUE OF X IS
    0.13025600 01        0.16971260 01
   THE CONSTRAINT VALUES
    0.46235060-03        0.31344850-03


  2ND  ORDER ESTIMATES
     F= 0.73666940 01     P= 0.73684970 01     G= 0.7366694D 01     RSIGMA= 0.0              H= 0.0
   THE CURRENT VALUE OF X IS
    0.13027750 01        0.16972240 01
   THE CONSTRAINT VALUES
   -0.27612520-06       -0.62938790-06
```

Figure 3.--*continued*

```
                                    P=  0.73669440 01      G=  0.73666940 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.13027750 01        0.16972240 01
 THE CONSTRAINT VALUES
  0.82491970-06        0.65044930-06


LAGRANGE MULTIPLIERS
     F=  0.73666950 01      P=  0.73689440 01      G=  0.73666940 01      RSIGMA=  0.0                H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027750 01        0.16972240 01
  THE CONSTRAINT VALUES
  0.13201060 01        0.19672150 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
  TIME=     1.371 SECONDS

**********************************************
(19)  POINT=  23       DOTT=  0.62757270-06      RHO=  0.15258790-03      MAGNITUDE=  0.20081460 00      PHASE= 2
     F=  0.73669930 01      P=  0.73698230 01      G=  0.73666880 01      RSIGMA=  0.28304970-02      H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027230 01        0.16972000 01
  THE CONSTRAINT VALUES
  0.11404160-03        0.77054420-04


2ND  ORDER ESTIMATES
     F=  0.73666860 01      P=  0.73671440 01      G=  0.73666860 01      RSIGMA=  0.0                H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027770 01        0.16972250 01
  THE CONSTRAINT VALUES
 -0.22661880-05       -0.19032050-05


1ST  ORDER ESTIMATES
     F=  0.73666860 01      P=  0.73672560 01      G=  0.73666860 01      RSIGMA=  0.0                H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027770 01        0.16972250 01
  THE CONSTRAINT VALUES
 -0.20730320-05       -0.17436010-05


LAGRANGE MULTIPLIERS
     F=  0.73666860 01      P=  0.73672560 01      G=  0.73666860 01      RSIGMA=  0.0                H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027770 01        0.16972250 01
  THE CONSTRAINT VALUES
  0.13380070 01        0.13302610 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
  TIME=     1.461 SECONDS

**********************************************
(20)  POINT=  25       DOTT=  0.48066420-07      RHO=  0.38146970-04      MAGNITUDE=  0.81424870-01      PHASE= 2
     F=  0.73667630 01      P=  0.73675810 01      G=  0.73666920 01      RSIGMA=  0.81243640-03      H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027620 01        0.16972180 01
  THE CONSTRAINT VALUES
  0.28860580-04        0.19511670-04


2ND  ORDER ESTIMATES
     F=  0.73666940 01      P=  0.73668050 01      G=  0.73666940 01      RSIGMA=  0.0                H=  0.0
  THE CURRENT VALUE OF X IS
  0.13027750 01        0.16972240 01
  THE CONSTRAINT VALUES
```

Figure 3.--*continued*

```
 1ST  ORDER ESTIMATES
    F=  0.7366940 01      P=  0.7366833D 01      G=  0.7366694D 01      RSIGMA=  0.0                H=  0.0
  THE CURRENT VALUE OF X IS
   0.1302775) 01         0.1697224D 01
  THE CONSTRAINT VALUES
   0.4661930)-06         0.3307564D-06


 LAGRANGE MULTIPLIERS
     F=  0.7366694D 01      P=  0.7366833D 01      G=  0.7366694D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302775) 01         0.1697224D 01
   THE CONSTRAINT VALUES
    0.1321768) 01         0.1955085D 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
     TIME=     1.621 SECONDS

**********************************************
(21)      POINT=  27        DOTT=  0.5385601D-07      RHO=  0.9536743D-05      MAGNITUDE=  0.2332791D 00      PHASE= 2
     F=  0.7366711D 01      P=  0.7366941D 01      G=  0.7366692D 01      RSIGMA=  0.2298500D-03      H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302772) 01         0.1697223D 01
   THE CONSTRAINT VALUES
    0.7118222D-05         0.4791486D-05


 2ND  ORDER ESTIMATES
    F=  0.7366692D 01      P=  0.7366721D 01      G=  0.7366692D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302776) 01         0.1697224D 01
   THE CONSTRAINT VALUES
   -0.1689720D-06        -0.1469764D-06


 1ST  ORDER ESTIMATES
    F=  0.7366692D 01      P=  0.7366728D 01      G=  0.7366692D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302776) 01         0.1697224D 01
   THE CONSTRAINT VALUES
   -0.1292741D-06        -0.1152431D-06


 LAGRANGE MULTIPLIERS
     F=  0.7366692D 01      P=  0.7366728D 01      G=  0.7366692D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302776) 01         0.1697224D 01
   THE CONSTRAINT VALUES
    0.1339765) 01         0.1990352D 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
     TIME=     1.711 SECONDS

**********************************************
(22)      POINT=  29        DOTT=  0.5868017D-07      RHO=  0.2384186D-05      MAGNITUDE=  0.7124525D 00      PHASE= 2
     F=  0.7366697) 01      P=  0.7366761D 01      G=  0.7366692D 01      RSIGMA=  0.6390440D-04      H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302775) 01         0.1697224D 01
   THE CONSTRAINT VALUES
    0.1826825)-05         0.1252313D-05


 1ST  ORDER ESTIMATES
    F=  0.7366693D 01      P=  0.7366699D 01      G=  0.7366693D 01      RSIGMA=  0.0                H=  0.0
   THE CURRENT VALUE OF X IS
    0.1302775) 01         0.1697224D 01
```

Figure 3.--*continued*

```
 .7334...-07          0.95110400-07

1.T  ORDER ESTIMATES
  F=  0.73666930 01      P=  0.73667010 01      G=  0.73666930 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.13027760 01         0.16972240 01
 THE CONSTRAINT VALUES
  0.63023410-07         0.72588310-07

LAGRANGE MULTIPLIERS
  F=  0.73666930 01      P=  0.73667010 01      G=  0.73666930 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.13027760 01         0.16972240 01
 THE CONSTRAINT VALUES
  0.13050980 01         0.19038260 01

                         SENSITIVITY ANALYSIS

(23)  THE VALUE OF R(RHO) IS 0.23842D-05

       THE POINT AT WHICH THE ESTIMATE OF SENSITIVITY WILLBE MADE IS
     X( 1)=  1.302775       X( 2)=  1.697224       X(

       PARAMETER VALUE    DIFFERENCING INTERVAL
   1     0.0                  0.10000D-09
   2    -3.00000              0.10000D-09


                  OPTIMAL VALUE FUNCTION SENSITIVITY

(24) DF/DA( 1)=  1.305098      DF/DA( 2)=  1.903824     DF/DA(
(25) DETAILED SENSITIVITY RESULTS FOLLOW FOR PARAMETERS
      1 , 2 ,

(26)  X-DERIVATIVES ARE WITH RESPECT TO PARAMETER , 1
     DX( 1)=-0.2773500      DX( 2)= 0.2773500      DX(
(27)  U-DERIVATIVES WITH RESPECT TO PARAMETER 1
     DU( 1)= 0.5063357      DU( 2)=-0.4622335D-01 DU(
(28)          DF(X(R))/DA=     1.32820
     *********

      X-DERIVATIVES ARE WITH RESPECT TO PARAMETER , 2
(29) DX( 1)=-0.2773497      DX( 2)=-0.7226484      DX(
      U-DERIVATIVES WITH RESPECT TO PARAMETER 2
     DU( 1)=-0.4622325D-01 DU( 2)=  1.388571       DU(
              DF(X(K))/DA=     1.93375
     *********
```

Figure 3.--*continued*

```
(II)
NONLINEAR PROGRAMMING ROUTINE-SUMT VERSION 4 03/01/73

     N= 2      M= 2      MZ= 0

(30)     MAX. TIME= 0.9000000E 03    R= 0.1000000E 02    RATIO= 0.4000000E 01      EPSILON= 0.9999997E-08    THETA= 0.9999996E-06

OPTIONS SELECTED
    3     2     1     1     1     1     1     0     1     1

TOLERANCES
     0.0                0.9999999E-03

SECOND SET OF OPTIONS
    1     1     1     1     1     0     1
    TIME=    0.012 SECONDS
      F=  0.92500000 01     P=  0.0              G=  0.0              RSIGMA=  0.0              H=  0.0
    THE CURRENT VALUE OF X IS
     0.10000000 01       0.15000000 01
    THE CONSTRAINT VALUES
     0.15000000 01       0.50000000 00
    TIME=    0.090 SECONDS

*****THE FEASIBLE STARTING POINT TO BE USED IS ...
      F=  0.92500000 01     P=  0.0              G=  0.0              RSIGMA=  0.0              H=  0.0
    THE CURRENT VALUE OF X IS
     0.10000000 01       0.15000000 01
    THE CONSTRAINT VALUES
     0.15000000 01       0.50000000 00
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.250 SECONDS

**********************************************
        POINT=   3       DOTT=  0.1649188D-02      RHO=  0.10000000 02      MAGNITUDE=  0.1642964D 00      PHASE= 2
      F=  0.15329420 02     P=  0.26702730 01     G= -0.46705840 01     RSIGMA= -0.12659140 02     H=  0.0
    THE CURRENT VALUE OF X IS
     0.16506330 00       0.12109010 01
    THE CONSTRAINT VALUES
     0.21836550 01       0.16240350 01


  LAGRANGE MULTIPLIERS
      F=  0.15329420 02     P=  0.26702730 01     G= -0.46705850 01     RSIGMA= -0.12659140 02     H=  0.0
    THE CURRENT VALUE OF X IS
     0.16506330 00       0.12109010 01
    THE CONSTRAINT VALUES
     0.45794770 01       0.61575010 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.449 SECONDS

**********************************************
        POINT=   5       DOTT=  0.7491886D-03      RHO=  0.25000000 01      MAGNITUDE=  0.7059479D-01      PHASE= 2
      F=  0.10077730 02     P=  0.96605790 01     G=  0.5077728D 01     RSIGMA= -0.4171488D 00     H=  0.0
    THE CURRENT VALUE OF X IS
     0.9053927D 00       0.12920920 01
    THE CONSTRAINT VALUES
     0.14723560 01       0.80251520 00


   1ST ORDER ESTIMATES
      F=  0.99736000 01     P=  0.11990690 02     G=  0.8327165D 01     RSIGMA=  0.0              H=  0.0
    THE CURRENT VALUE OF X IS
     0.11521670 01       0.13191560 01
    THE CONSTRAINT VALUES
```

Figure 3.--*continued*

```
1ST  ORDER ESTIMATES
   F=  0.62613650 01      P=  0.62615070 01      G=  0.62613650 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615530 01          0.14384470 01
 THE CONSTRAINT VALUES
-0.75033150-06          -0.35286570-06


LAGRANGE MULTIPLIERS
   F=  0.62613650 01      P=  0.62615070 01      G=  0.62613650 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615530 01          0.14384470 01
 THE CONSTRAINT VALUES
  0.91698530 00          0.20602070 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
 TIME=     2.391 SECONDS

**********************************************
      POINT=  23        DOTT=  0.90250670-07      RHO=  0.95367430-05      MAGNITUDE=  0.31949610 00      PHASE= 2
   F=  0.62613850 01      P=  0.62616120 01      G=  0.62613660 01      RSIGMA=  0.22689850-03      H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615490 01          0.14384460 01
 THE CONSTRAINT VALUES
  0.10230960-04          0.45629360-05


2ND  ORDER ESTIMATES
   F=  0.62613660 01      P=  0.62613940 01      G=  0.62613660 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615530 01          0.14384470 01
 THE CONSTRAINT VALUES
-0.19056200-06          -0.98911230-07


1ST  ORDER ESTIMATES
   F=  0.62613660 01      P=  0.62614020 01      G=  0.62613660 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615530 01          0.14384470 01
 THE CONSTRAINT VALUES
-0.22557890-06          -0.11478360-06


LAGRANGE MULTIPLIERS
   F=  0.62613660 01      P=  0.62614020 01      G=  0.62613660 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615530 01          0.14384470 01
 THE CONSTRAINT VALUES
  0.93214540 00          0.20992480 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
 TIME=     2.500 SECONDS

**********************************************
      POINT=  25        DOTT=  0.29651740-07      RHO=  0.23841860-05      MAGNITUDE=  0.42775980 00      PHASE= 2
   F=  0.62613710 01      P=  0.62614340 01      G=  0.62613660 01      RSIGMA=  0.63127680-04      H=  0.0
 THE CURRENT VALUE OF X IS
  0.15615520 01          0.14384470 01
 THE CONSTRAINT VALUES
  0.26639090-05          0.11895300-05


2ND  ORDER ESTIMATES
   F=  0.62613670 01      P=  0.62613730 01      G=  0.62613670 01      RSIGMA=  0.0                H=  0.0
 THE CURRENT VALUE OF X IS
```

(31)

Figure 3.--*continued*

```
 THE CONSTRAINT VALUES
  0.15503[illegible]D-06        0.8916[illegible]D-07


 1ST ORDER ESTIMATES
    F=  0.62613670 01     P=  0.62613750 01     G=  0.62613670 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.15615530 01        0.14384470 01
  THE CONSTRAINT VALUES
   0.14155530-06        0.71729250-07


LAGRANGE MULTIPLIERS
     F=  0.62613670 01     P=  0.62613750 01     G=  0.62613670 01     RSIGMA=  0.0              H=  0.0
  THE CURRENT VALUE OF X IS
   0.15615530 01        0.14384470 01
  THE CONSTRAINT VALUES
   0.89499520 00        0.20043080 01
```

(32)

```
                    SENSITIVITY ANALYSIS

 THE VALUE OF R(RHO) IS 0.23842D-05

    THE POINT AT WHICH THE ESTIMATE OF SENSITIVITY WILLBE MADE IS
X( 1)=  1.561552      X( 2)=  1.438447      X(

    PARAMETER VALUE   DIFFERENCING INTERVAL
 1    -1.00000             0.100000-09
 2    -3.00000             0.100000-09


                OPTIMAL VALUE FUNCTION SENSITIVITY
```

(33)

```
DF/DA( 1)= 0.8949943     DF/DA( 2)=  2.004307     DF/DA(
```


Figure 3.--*continued*

III

OPTIMAL VALUE FUNCTION BOUNDS WHEN PAR(1) IS PERTURBED

POINT 1 (UNPERTURBED SOLUTION) :
PAR(1)= 0.2931668D-15 F(PAR(1))= 0.73666930 01

POINT 2 (PERTURBED SOLUTION) :
PAR(1))= -0.10000000 01 F(PAR(1)= 0.62613670 01

LINE PASSING THROUGH POINTS 1 AND 2 AND OVER ESTIMATING F*

F = 0.11053260 01 * PAR(1) + 0.73666930 01

LINE UNDER ESTIMATING F* AT POINT 1

F = 0.13350990 01 * PAR(1) + 0.73666930 01

LINE UNDER ESTIMATING F* AT POINT 2

F = 0.89499430 00 * PAR(1) + 0.71563610 01

QUADRATIC ESTIMATION OF F* THROUGH POINTS 1 AND 2

F = 0.19977190 00 * PAR(1)**2 + 0.13050980 01 * PAR(1) + 0.73666930 01

F* BOUND EVALUATION AT TEN EQUIDISTANCE POINTS BETWEEN POINTS 1 AND 2

PAR(1)	LOWER BOUND	UPPER BOUND	QUAD ESTIMATE
0.000	0.73466930 01	0.73666930 01	0.73666930 01
-0.100	0.72361830 01	0.72561600 01	0.72381800 01
-0.200	0.71056730 01	0.71456270 01	0.71136640 01
-0.300	0.69751630 01	0.70350950 01	0.69931430 01
-0.400	0.68446530 01	0.69245620 01	0.68766170 01
-0.500	0.67141440 01	0.68140300 01	0.67640870 01
-0.600	0.66193640 01	0.67034970 01	0.66555520 01
-0.700	0.65298650 01	0.65929640 01	0.65510120 01
-0.800	0.64403650 01	0.64824320 01	0.64504680 01
-0.900	0.63508660 01	0.63718990 01	0.63539200 01
-1.000	0.62613670 01	0.62613670 01	0.62613670 01

Figure 3.--*continued*

```
NONLINEAR PROGRAMMING ROUTINE-SUMT VERSION 4 03/01/73

    M= 2      N= 2      MZ= 0

  MAX. TIME= 0.9000000E 03    R= 0.1000000E 02    RATIO= 0.4000000E 01    EPSILON= 0.9999997E-08    THETA= 0.9999996E-06

OPTIONS SELECTED
    3     2     1     1     1     1     1     0     1     1

TOLERANCES
    0.0                0.9999999E-03

SECOND SET OF OPTIONS
    1     1     1     1     1     0     1
    TIME=    0.012 SECONDS
     F= 0.92500000 01     P= 0.0              G= 0.0              RSIGMA= 0.0              H= 0.0
    THE CURRENT VALUE OF X IS
     0.10000000 01       0.15000000 01
    THE CONSTRAINT VALUES
     0.50000000 00      -0.15000000 01
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.219 SECONDS

**********************************************
        POINT=   3       DOTT=  0.14928790-03      RHO=  0.10000000 02      MAGNITUDE=  0.12154220-01      PHASE= 1
     F=  0.87508970 01     P= -0.14275850 02     G= -0.11249100 02     RSIGMA= -0.23026750 02     H=  0.0
    THE CURRENT VALUE OF X IS
    -0.50030240 00       0.10251200 02
    THE CONSTRAINT VALUES
     0.10000900 02      -0.87508970 01


 LAGRANGE MULTIPLIERS
     F=  0.87508970 01     P= -0.14275850 02     G= -0.11249100 02     RSIGMA= -0.23026750 02     H=  0.0
    THE CURRENT VALUE OF X IS
    -0.50030240 00       0.10251200 02
    THE CONSTRAINT VALUES
     0.99991030 00      -0.11427400 01
    TIME=    0.379 SECONDS

*****THE FEASIBLE STARTING POINT TO BE USED IS ...
     F=  0.20504440 02     P=  0.12791100 02     G=  0.0              RSIGMA=  0.12791100 02     H=  0.0
    THE CURRENT VALUE OF X IS
    -0.49994940 00       0.14951330 01
    THE CONSTRAINT VALUES
     0.12456840 01       0.48164300-02
APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DETERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.
    TIME=    0.660 SECONDS

**********************************************
        POINT=   7       DOTT=  0.89618260-05      RHO=  0.10000000 02      MAGNITUDE=  0.16890250-01      PHASE= 2
     F=  0.20386780 02     P=  0.30183540 02     G=  0.86778500-01     RSIGMA=  0.10096760 02     H=  0.0
    THE CURRENT VALUE OF X IS
    -0.50167520 00       0.74207750 00
    THE CONSTRAINT VALUES
     0.65106950 00       0.55959770 00


 LAGRANGE MULTIPLIERS
     F=  0.20086780 02     P=  0.30183540 02     G=  0.86778500-01     RSIGMA=  0.10096760 02     H=  0.0
    THE CURRENT VALUE OF X IS
    -0.50167520 00       0.74207750 00
    THE CONSTRAINT VALUES
     0.15459360 01       0.17869980 02
```

Figure 3.--*continued*

V

OPTIMAL VALUE FUNCTION BOUNDS WHEN PAR(2) IS PERTURBED

POINT 1 (UNPERTURBED SOLUTION) :
PAR(2)= -0.30000000 01 F(PAR(2))= 0.73666930 01

POINT 2 (PERTURBED SOLUTION) :
PAR(2)= -0.10000000 01 F(PAR(2)= 0.14055730 02

LINE PASSING THROUGH POINTS 1 AND 2 AND OVER ESTIMATING F*

F = 0.33445180 01 * PAR(2) + 0.17400250 02

LINE UNDER ESTIMATING F* AT POINT 1

F = 0.19038240 01 * PAR(2) + 0.13078170 02

LINE UNDER ESTIMATING F* AT POINT 2

F = 0.47961840 01 * PAR(2) + 0.18851910 02

QUADRATIC ESTIMATION OF F* THROUGH POINTS 1 AND 2

F = 0.72034670 00 * PAR(2)**2 + 0.62259050 01 * PAR(2) + 0.19561290 02

F* BOUND EVALUATION AT TEN EQUIDISTANCE POINTS BETWEEN POINTS 1 AND 2

PAR(2)	LOWER BOUND	UPPER BOUND	QUAD ESTIMATE
-3.000	0.73666930 01	0.73666930 01	0.73666930 01
-2.900	0.77474570 01	0.80355960 01	0.77762710 01
-2.800	0.81282220 01	0.87045000 01	0.82434780 01
-2.700	0.85089870 01	0.93734030 01	0.87683120 01
-2.600	0.88897520 01	0.10042310 02	0.93507740 01
-2.500	0.92705170 01	0.10711210 02	0.99908640 01
-2.400	0.96512810 01	0.11380110 02	0.10688540 02
-2.300	0.10032050 02	0.12049020 02	0.11443930 02
-2.200	0.10412810 02	0.12717920 02	0.12256900 02
-2.100	0.10793580 02	0.13386820 02	0.13127500 02
-2.000	0.11174340 02	0.14055730 02	0.14055730 02

T-434

Figure 3.--*continued*

I Output for solution of the unperturbed problem, i.e., $\varepsilon_1 = 0$, $\varepsilon_2 = -3$ (Step 1 of Algorithm 3.1).

(1) Printout of parameter card data.

(2) Printout of first option card data.

(3) Printout of tolerance card data.

(4) Printout of second option card data.

(5) Elapsed time since the start of the problem. (This information is not accurate. Subroutine TIMEC, which monitors the elapsed time, has to be modified.)

(6) Initial starting point x^0 and corresponding problem function values.

(7) Feasibility of x^0 is verified. Note that if x^0 is not feasible, then SENSUMT invokes the subroutines FEAS and BODY to find a feasible starting point.

(8) The solution to the first subproblem. Program took four inverse product moves to minimize W for

RHØ = 10.,(r=10)

$$\text{DØTT} = \nabla_x W[x(r)]^T \left[\frac{\partial^2 W[x(r)]}{\partial x_i \partial x_j}\right]^{-1} \nabla_x W[x(r)] < 10^{-7}$$

MAGNITUDE = $||\nabla_x W[x(r)]||$ (the magnitude of the gradient of W)

F = f[x(r)] (the value of the objective function)

$$W = f[x(r)] - r \sum_{j=1} \ln g_j[x(r)] + \sum_{j=M+1}^{M+MZ} \left[h_j[x(r)]\right]^2 / r$$

(the value of the P-function)

Note: The W-function coded in SENSUMT differs from that given in (2.6) in the last term by a factor of 2.

$$\text{RESIGMA} = -r \sum_{j=1}^{M} \ell n\, g_j[x(r)] \quad \text{(note that this may be negative)}$$

$$H = \sum_{j=M+1}^{M+MZ} h_j^2[x(r)]/r$$

G = dual value = f[x(r)] - r*M + 2.*H

(in a convex problem $F \geq v^* \geq G$)

where the current value of x is x(r) and the current constraint values are $g_1[(r)]$ and $g_2[x(r)]$.

(9) Values of F, W (designated by P in the computer output), G, RSIGMA, and x are repeated. Current constraint values (i.e., u_j and w_j) are r/g_j , j=1,M , and $2n_j(x)/r$, j=M+1,M+MZ. Note: Since Example 1 has no equality constraint, the entries relating to h_j in 8 and 9 are zero.

(10) The solution to the second subproblem (r = 10/4 = 2.5).

(11) The current first order estimates of x (obtained by first order estrapolation, using the solution of the first and second subproblems) and the corresponding problem functions.

(12) The solution to the third subproblem (r = 2.5/4 = .625).

(13) The current second order estimates of x (obtained by second order extrapolation using solutions of the first, second, and third subproblems) and the corresponding problem functions.

(14) The solution to the fourth subproblem.

(15) The solution to the fifth subproblem.

(16) The solution to the sixth subproblem.

(17) The solution to the seventh subproblem.

(18) The solution to the eighth subproblem.

(19) The solution to the ninth subproblem.

(20) The solution to the tenth subproblem.

(21) The solution to the eleventh subproblem.

(22) The solution to the twelvth (final) subproblem.

(23) The general information about the sensitivity analysis including the value of r , the estimated final solution point and the parameter values and associated differencing intervals.

(24) Estimates of the sensitivity of the optimal value function obtained by taking the gradient of the Lagrangian with respect to the parameters as described in Step 9 of Algorithm 2.1.

(25) The parameters for which detailed sensitivity results follow. Here it is both parameters since the optimal value function is sensitive to both.

(26) Estimates of the partial derivatives of the solution point taken with respect to parameter one as in Steps 1-3 of Algorithm 2.1.

(27) Estimates of the partial derivatives of the Lagrange multipliers taken with respect to parameter one as in Step 5 of Algorithm 2.1.

(28) Estimate of the partial derivative of the optimal value function taken with respect to parameter one and obtained by using the chain rule as in Step 8 of Algorithm 2.1.

(29) Same as (26), (27), and (28) but with respect to parameter two.

II Output for solution of the first perturbed problem, i.e., $\varepsilon_1 = -1$, $\varepsilon_2 = -3$ (Step 2 of Algorithm 3.1).

(30) Printout of the input data and subproblem solutions for the first perturbed problem.

(31) Printout of the final subproblem for the first perturbed problem.

(32) Same as (23) but for the first perturbed problem.

(33) Same as (24) but for the first perburbed problem.

III Output for optimal value function bounds as a function of first parameter (Steps 3 and 4 of Algorithm 3.1).

(34) $(f^*(\varepsilon_1),\varepsilon_1)$ of the unperturbed and first perturbed problems [see (22), (23), and (31), (32), respectively].

(35) Equations of parametric upper bound on $f^*(\varepsilon_1)$.

(36) Equations of parametric lower bounds on $f^*(\varepsilon_1)$.

(37) Maximum of these equations over the range of $-1 \leq \varepsilon_1 \leq 0$ provides a lower bound on $f^*(\varepsilon_1)$.

(38) Equation of a quadratic estimate of $f^*(\varepsilon_1)$ [see the description of subroutine BOUND in Section 6 for the method of deriving this equation].

(39) Value of bounds and the quadratic estimate of $f^*(\varepsilon_1)$ at eleven equidistant points over the perturbation range of the first parameter ε_1.

IV Output for solution of the second perturbed problem, i.e., $\varepsilon_1 = 0$, $\varepsilon_2 = -1$ (Step 2 of Algorithm 3.1).

Description of the output for the second perturbed problem is similar to that of the first perturbed problem, i.e., (30) - (33).

V Output for optimal value function bounds as a function of the second parameter (Steps 3 and 4 of Algorithm 3.1).

The description here again closely parallels that of the first parameter, i.e., (34) - (39).

The graphical depiction of the bounds derived for $f^*(\varepsilon)$ as a function of ε_2, together with a plot of the analytical solution and quadratic estimates of $f^*(\varepsilon)$ at different values of ε_2 is in Figure 4.

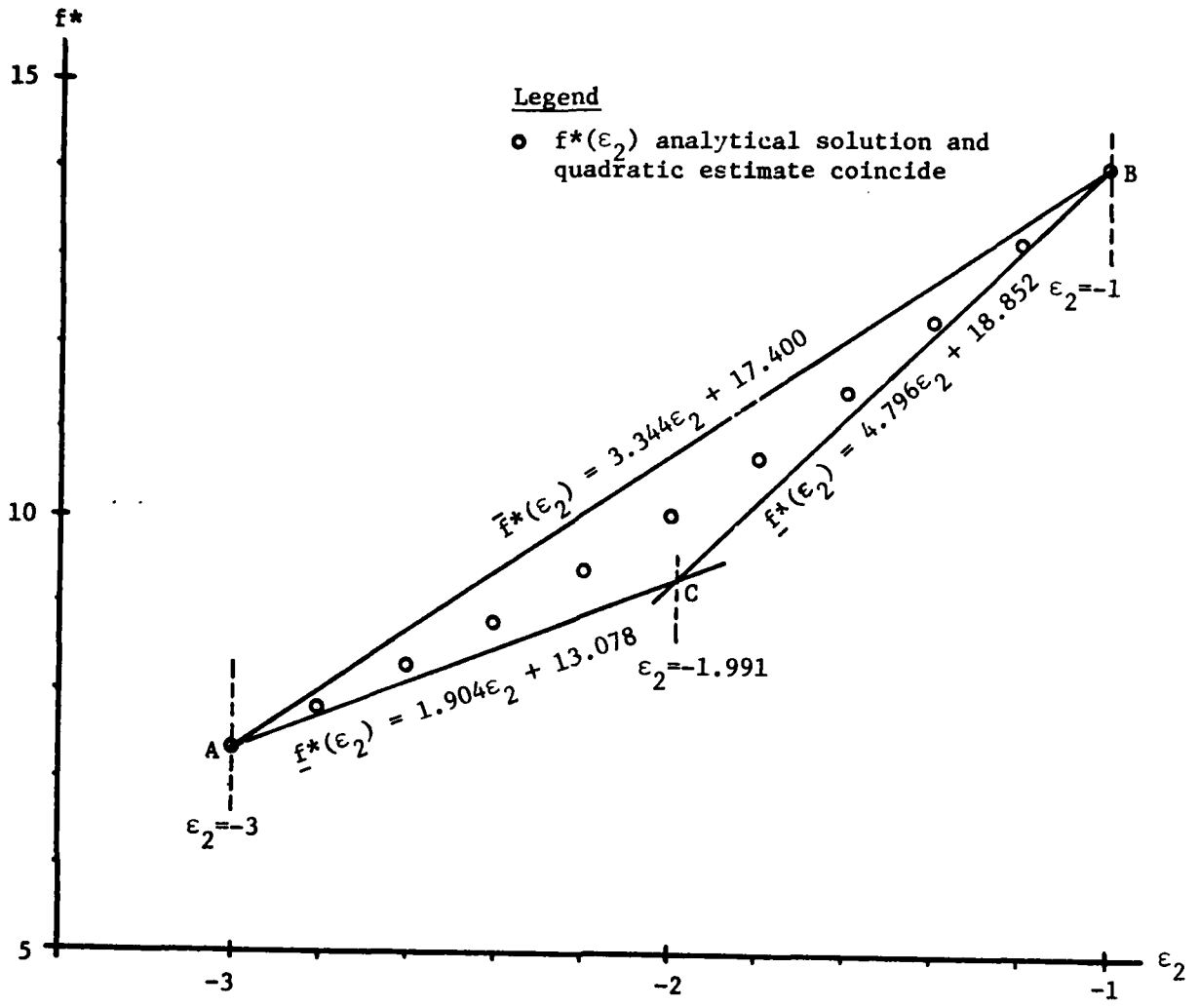


Figure 4.--Graph of bounds on $f^*(\varepsilon_2)$ of Example 1 (computer solution).

5.2 Example 2

$$\underset{x_1,\ldots,x_5}{\text{Minimize}} \quad f(x) = x_1^2 + .5e^{x_2} - .125x_3^3 + .25(x_4-40)^2 - .1e^{x_5}$$

$$\text{Subject to} \quad g_1(x) = -.5x_1^2 + 6x_2 - 5e^{x_3} - .05x_4^2 - .5/x_5 \geq \varepsilon_1$$

$$g_2(x) = -5e^{-x_1} - 2x_2^2 + 3x_3 + x_4 + 3x_5 \geq -12$$

$$g_3(x) = 3x_1 + x_2 - x_3^2 + x_4 - x_5^2 \geq 2$$

$$0 \leq x_3 \leq 5 , \quad 0 \leq x_5 \leq 5 , \quad 0 \leq x_j \leq \infty , \quad j=1,2,4 .$$

It is desired to calculate bounds on $f^*(\varepsilon_1)$ of this problem for $\varepsilon_1 \in [-10,5]$.

(i) Problem SR(ε).

• Standard format:

$$\underset{x_1,\ldots,x_5}{\text{Min}} \quad f(x) = \sum_{j=1}^{5} f_j(x_j) = x_1^2 + .5e^{x_2} - .125x_3^3 + .25(x_4-40)^2 - .1e^{x_5}$$

S.t. $\quad x \in G \cap B$, where

$$G = \{x : g_1(x) \geq \varepsilon_1 \ , \ g_2(x) \geq -12 \ , \ g_3(x) \geq 2\} \ , \quad -10 \leq \varepsilon_1 \leq 5$$

and

$$B = \{0 \leq x_3 \leq 5 \ , \ 0 \leq x_5 \leq 5 \ , \ 0 \leq x_i \leq \infty \ , \ i=1,2,4\} \ .$$

(ii) Problem $\tilde{CSR}(\varepsilon)$, (convex overestimating problem of SR(ε)).

• Problem formulation:

$$\underset{\approx}{f}_3(x_3) = -3.125x_3 + 6.014$$

$$\underset{\approx}{f}_5(x_5) = -2.948x_5 + 7.027$$

Remaining terms are identical to those of problem SR(ε); thus problem $\tilde{CSR}(\varepsilon)$ reads

$$\underset{x_1,\ldots,x_5}{\text{Min}} \quad \underset{\approx}{f}(x) = x_1^2 + .5e^{x_2} - 3.125x_3 + .25(x_4-40)^2 - 2.948x_5 + 13.041$$

S.t. $\quad x \in \tilde{\tilde{G}} \cap B \ , \quad -10 \leq \varepsilon_1 \leq 5 \ ,$

where $\quad \tilde{\tilde{G}} \equiv G \ .$

Note: $\underset{\approx}{f}_3(x_3)$ and $\underset{\approx}{f}_5(x_5)$ were chosen here to be the lowest nonunderestimating lines parallel to the convex envelopes of $f_3(x_3)$ and $f_5(x_5)$, respectively.

(iii) Problem $\underset{\sim}{CSR}(\varepsilon)$, (convex underestimating problem).

• Problem formulation:

$$\underset{\sim}{f}_3(x_3) = -3.125x_3$$

$$\underset{\sim}{f}_5(x_5) = -2.948x_5$$

Remaining terms are identical to those of problem $SR(\varepsilon)$; thus problem $\underset{\sim}{CSR}(\varepsilon)$ reads

$$\underset{x_1,\ldots,x_5}{\text{Min}} \quad f(x) = x_1^2 + .5e^{x_2} - 3.12x_3 + .25(x_4-40)^2 - 2.948x_5 - .1$$

$$\text{S.t.} \quad x \in \tilde{G} \cap B \ , \quad -10 \leq \varepsilon_1 \leq 5$$

where $\tilde{G} \equiv G$.

The computer listing of the code for problem $\tilde{CSR}(\varepsilon)$ is shown in Figure 5. The listing of the code for $\underset{\sim}{CSR}(\varepsilon)$ is identical to that of the problem $\tilde{CSR}(\varepsilon)$ except for the constant term in the objective function, which is + 13.041 , rather than - .1 .

(iv) Bounds.

Figures 6 and 7 depict the calculated upper and lower bounds via the analysis of the problems $\tilde{CSR}(\varepsilon)$ and $\underset{\sim}{CSR}(\varepsilon)$, respectively. These results are summarized below.

Upper bound:

$$\bar{f}^*(\varepsilon_1) = 4.825\varepsilon_1 + 136.628 \ .$$

Lower bound:

$$\underline{f}^*(\varepsilon_1) = \max[1.880\varepsilon_1 + 94.032 \ , \ 5.466\varepsilon_1 + 120.284] \ .$$

The graphical depiction of these bounds is in Figure 8.

```
J=GnAEMI  T=2  L=2
// EXEC FORGI
      SUBROUTINE READIN
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      COMMON/ABG/LY,LZ,PER(45)
      NPAR=3
      PAR(1)=5.
      PAR(2)=-12.
      PAR(3)=2.
      PER(1)=-15.
      RETURN
      END
      SUBROUTINE RESINI(IN,VAL)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SHARE/X(45),DEL(45), A(45,45),N,M,MN,NP1,NM1
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      COMMON/ABG/LY,LZ,PER(45)
      I=IN+1
      GO TO (20,1,2,3,4,5),I
   20 VAL=X(1)**2+.5*DEXP(X(2))-3.125*X(3)+.25*(X(4)-40.)**2
     * -2.948*X(5)+13.041
      RETURN
    1 VAL=-.5*X(1)**2+6.*X(2)-5.*DEXP(X(3))-.05*X(4)**2-.5/X(5)
     * -PAR(1)
      RETURN
    2 VAL=-5.*DEXP(-X(1))-2.*X(2)**2+3.*X(3)+X(4)+3.*X(5)-PAR(2)
      RETURN
    3 VAL=3.*X(1)+X(2)-X(3)**2+X(4)-X(5)**2-PAR(3)
      RETURN
    4 VAL=-X(3)+5.
      RETURN
    5 VAL=-X(5)+5.
      RETURN
      END
      SUBROUTINE GRADI(IN)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SHARE/X(45),DEL(45), A(45,45),N,M,MN,NP1,NM1
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      J=IN+1
      DO 15 I=1,N
      DEL(I)=0.0
   15 CONTINUE
      GO TO (20,1,2,3,4,5),J
   20 DEL(1)=2.*X(1)
      DEL(2)=.5*DEXP(X(2))
      DEL(3)=-3.125
      DEL(4)=.5*(X(4)-40.)
      DEL(5)=-2.948
      RETURN
    1 DEL(1)=-1.*X(1)
      DEL(2)=6.
      DEL(3)=-5.*DEXP(X(3))
      DEL(4)=-.1*X(4)
      DEL(5)=.5/X(5)**2
      RETURN
    2 DEL(1)=5.*DEXP(-X(1))
      DEL(2)=-4.*X(2)
      DEL(3)=3.
      DEL(4)=1.
      DEL(5)=3.
```

Figure 5.--Computer listing of the code for the convex overestimating problem of Example 2.

```
      RETURN
    3 DEL(1)=3.
      DEL(2)=1.
      DEL(3)=-2.*X(3)
      DEL(4)=1.
      DEL(5)=-2.*X(5)
      RETURN
    4 DEL(3)=-1.
      RETURN
    5 DEL(5)=-1.
      RETURN
      END
      SUBROUTINE MATRIX(IN,K)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SHARE/X(45),DEL(45), A(45,45),N,M,MN,NP1,NM1
      COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS
      L=IN+1
      GO TO (20,1,2,3,4,4),L
   20 A(1,1)=2.
      A(2,2)=.5*DEXP(X(2))
      A(4,4)=.5
      RETURN
    1 A(1,1)=-1.
      A(3,3)=-5.*DEXP(X(3))
      A(4,4)=-.1
      A(5,5)=-1./X(5)**3
      RETURN
    2 A(1,1)=-5.*DEXP(-X(1))
      A(2,2)=-4.
      RETURN
    3 A(3,3)=-2.
      A(5,5)=-2.
      RETURN
    4 K=1
      RETURN
      END
// EXEC FORG6,DSN='OR799661.BOUNDS45',PROG=MAIN
//GO.SYSLIB  DD
//           DD
//  DD  DSN=GWU.FORTLIB,DISP=SHR
//GO.SYSLIN  DD
//  DD  DSN=&TEMPUNCH,DISP=(OLD,DELETE)
//F107F001  DD  SYSOUT=A
     1.0E-08        10.0      1.0E-06          4.0        900.0   5   5   0
        3.          3.          4.          1.          4.
      3       1       1       1       1       1               1       1
    0.1E-08      0.0001
      1       4       1       1       1       0       2
//
```

Figure 5.--*continued*

OPTIMAL VALUE FUNCTION BOUNDS WHEN PAR(1) IS PERTURBED
****

POINT 1 (UNPERTURBED SOLUTION) :
PAR(1)= 0.5000000 01 F(PAR(1))= 0.1607547D 03

POINT 2 (PERTURBED SOLUTION) :
PAR(1))= -0.1000000D 02 F(PAR(1)= 0.9837377D 02

LINE PASSING THROUGH POINTS 1 AND 2 AND OVER ESTIMATING F*

F = -0.4825398D 01 * PAR(1) + 0.1366277D 03

F* BOUND EVALUATION AT TEN EQUIDISTANCE POINTS BETWEEN POINTS 1 AND 2

PAR(1)	LOWER BOUND	UPPER BOUND	QUAD ESTIMATE
5.000		0.1607547D 03	
3.500		0.1535166D 03	
2.000		0.1462785D 03	
0.500		0.1390404D 03	
-1.000		0.1318023D 03	
-2.500		0.1245643D 03	
-4.000		0.1173262D 03	
-5.500		0.1100881D 03	
-7.000		0.1028500D 03	
-8.500		0.9561187D 02	
-10.000		0.9837377D 02	

Figure 6.--Parametric upper bound on $f^*(\varepsilon_1)$ via problem $\tilde{CSR}(\varepsilon)$, Example 2 (computer solution).

OPTIMAL VALUE FUNCTION BOUNDS WHEN PARM 11 IS PERTURBED
****

POINT 1 (UNPERTURBED SOLUTION) :
PARM 11= 0.50000000 01 F(PARM 11)= 0.14761370 03

POINT 2 (PERTURBED SOLUTION) :
PARM 11)= -0.10000000 02 F(PARM 11)= 0.75232770 02

LINE UNDER ESTIMATING F* AT POINT 1

F = 0.94659210 01 * PARM 11) + 0.12028410 03

LINE UNDER ESTIMATING F* AT POINT 2

F = 0.18799810 01 * PARM 11) + 0.94032580 02

F* QUAD EVALUATION AT TEN EQUIDISTANCE POINTS BETWEEN POINTS 1 AND 2

PARM 11	LOWER BOUND	UPPER BOUND	QUAD ESTIMATE
5.000	0.14761370 03		
4.500	0.13941490 03		
4.000	0.13121600 03		
3.500	0.12301710 03		
-1.000	0.11481820 03		
-2.500	0.10661930 03		
-4.000	0.99420460 02		
-5.500	0.90221580 02		
-7.000	0.82022700 02		
-8.500	0.73823740 02		
-10.000	0.75232770 02		

Figure 7.--Parametric lower bound on $f^*(\varepsilon_1)$ via problem $\underset{\sim}{CSR}(\varepsilon)$, Example 2 (computer solution).

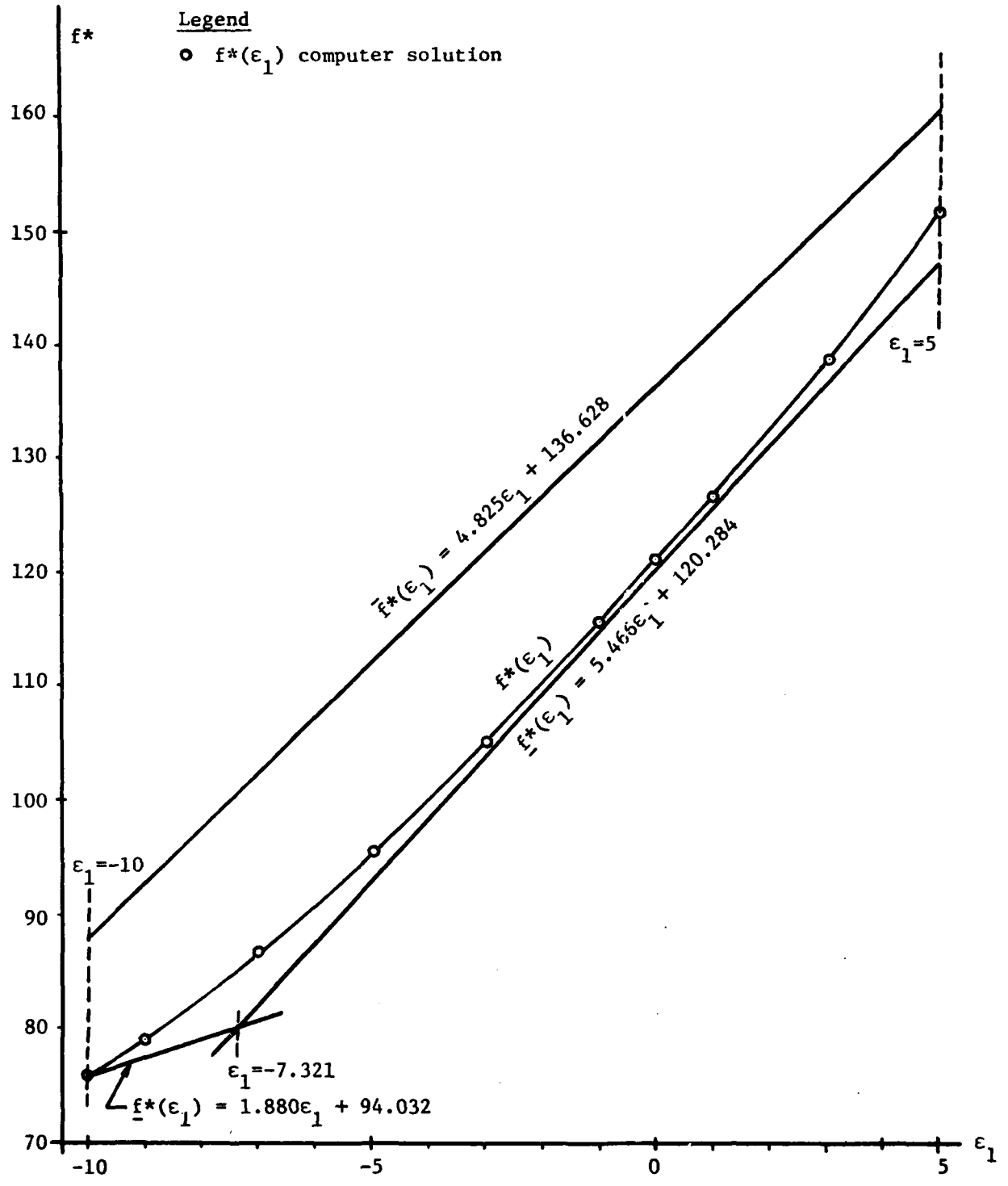


Figure 8.--Parametric bounds on $f^*(\varepsilon)$, Example 2 (computer solution).

6. General Description and Listing of the SENSUMT Subroutines

The subroutines comprising SENSUMT fall into four categories:

(i) user subroutines,
(ii) SUMT subroutines,
(iii) sensitivity subroutines, and
(iv) bound subroutines.

As mentioned before, user subroutines are generally composed of four subroutines, i.e., READIN, RSTNT, GRAD1, and MATRIX. These subroutines respectively provide pertinent information about the problem, functions of the problem, their gradients and Hessian matrices. A detailed description of these subroutines, together with the corresponding codes for Examples 1 and 2, were given in Section 4 and Figures 2 and 5.

SUMT subroutines implement the Sequential Unconstrained Minimization Technique of Fiacco and McCormick [10]. The subroutines, along with program MAIN, comprising this group are:

- BØDY
- CHCKER
- CØNVRG
- DIFF1
- DIFF2
- ESTIM
- EVALU
- FEAS
- FINAL
- GRAD
- INVERS
- MAIN
- ØPT
- ØUTPUT
- PEVALU
- PUNCH
- REJECT
- RHØCØM
- SECØND
- SECØRD
- SET
- STØRE
- TCHECK
- TIMEC
- TRANS
- XMØVE

With the exception of the subroutine TRANS, the above routines have been developed by Mylander, Holmes, and McCormick [13]. Some of

these routines, in particular program MAIN and subroutines BØDY and ØUTPUT, have been modified for implementing sensitivity and bound calculation routines. Subroutine TRANS was recently developed to aid the user when analyzing a convex equivalent of geometric programming problems by SENSUMT. See page 104 for a more detailed description of this subroutine.

Sensitivity subroutines implement the sensitivity analysis Algorithm 2.1 and interface it with the SUMT subroutines. A brief history of the development of these subroutines was given in Section 1. The latest version of the codes comprising this group, due to Armacost [1], are subroutines SENS, PARDIF, LMULT, and PRESEN. Subroutines SENS and PRESEN have been slightly modified in implementing the bound calculation routines.

Bound subroutines, developed in [11], implement the bound calculation Algorithms 3.1 and 3.2 given in Section 3. The two subroutines in this group are BOUND and PERT.

All of the subroutines comprising SENSUMT are dimensioned to solve problems having at most 45 variables, 45 parameters, and 200 constraints. However, if the computer capacity permits, they may readily be re-dimensioned to solve problems of larger size. All of these subroutines are separately filed in the Conversational Monitor System (CMS) component of the IBM Virtual Machine facility 370 (VM/370) at The George Washington University Center for Academic and Administrative Computing under their corresponding names.

To make this manual self-contained, the computer listing and a general description of each subroutine is provided (in alphabetical order by name). The descriptions of the SUMT subroutines are largely taken from [13], and those of the sensitivity subroutines are taken from [1]. The user subroutines are problem-dependent, thus they are included in the following listings. For familiarity with the user subroutines, the reader should refer to the coding of Examples 1 and 2 provided in Figures 2 and 5, respectively.

6.1 BØDY

Subroutine BØDY coordinates the flow among the subroutines that actually do the calculations required by the various phases of the algorithm. The flow in this routine is slightly different when the program is in the feasibility phase (solving the entry problem) rather than the normal phase (solving the stated NLP problem). The listing appears as Figure 9.

6.2 BØUND

The subroutine BØUND, called by the program MAIN, generates the equations for the upper and lower bounds of the optimal value function as functions of the problem parameters. When analyzing a problem which is known to have a convex or concave optimal value function, BØUND also derives the equation for a quadratic function which approximates the optimal value function. If $f^*(\varepsilon)$ is convex (concave), it corresponds to the higher (lower) of the two quadratic functions, in the perturbation range of the parameter of interest, which passes through the two calculated points in $f^*(\varepsilon),\varepsilon$ space and has the calculated slope at each of these points. Evaluation and printout of this quadratic function (when applicable), and the upper and/or lower bound at eleven equidistant points over the perturbation range of the parameters of interest are also programmed in subroutine BØUND. The listing appears as Figure 10.

6.3 CHCKER

Subroutine CHCKER evaluates the first partial derivatives for all the functions at the starting point first by calling the user-supplied subroutine GRAD1 and then by calling DIFF1. The results are printed out to aid the user in debugging the subroutines used to describe the NLP problem he wishes to solve.

The matrix of second partial derivatives of each function is also evaluated by the two methods; first by calling MATRIX and then calling

```
FORTRAN IV G LEVEL 21          BODY          DATE = 80240          10/05/54

0001                SUBROUTINE BODY                                          80000040
                    C                                                        80000070
                    C              AUGUST 1971                               80000080
                    C                                                        80000090
                    C BODY COORDINATES THE FLOW AMONG THE SUBROUTINES THAT ACTUALLY DO THE  80000100
                    C CALCULATIONS REQUIRED BY THE VARIOUS PARTS OF THE ALGORITHM.          80000110
0002                     IMPLICIT REAL*8(A-H,O-Z)                            80000120
0003                     REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2            80000130
0004                     COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1  80000140
0005                     COMMON /CPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10  80000150
0006                     COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO               80000160
0007                     COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,  80000170
                        1RSIGI,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,  80000180
                        2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),          80000190
                        3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS         80000200
0008                     COMMON /CONPAR/ NF1,NF2,NF3                         80000210
0009                     COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2  80000220
0010                     COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS              80000230
0011                     COMMON/ABG/LY,LZ,PER(45)                            80000240
0012                     COMMON/ABG1/FE1,FE2                                 80000250
0013                     COMMON/ABG2/DF1(45),DF2(45)                         80000260
0014                     NF2=2                                               80000270
0015                     NF3=2                                               80000280
0016                     MN=0                                                80000290
0017                     NUMINI=0                                            80000300
0018                     ISENS = 0                                           80000310
                    C    OPTION OF GETTING INITIAL RHO                       80000320
0019                     CALL RHOCOM                                         80000330
0020                     CALL EVALU                                          80000340
0021                10   CALL XMOVE                                          80000350
0022                     GO TO (30,20), NT3                                  80000360
0023                20   CALL TIMEC                                          80000370
0024                     CALL OUTPUT (1)                                     80000380
0025                     GO TO 40                                            80000390
0026                30   CALL TCHECK                                         80000400
                    C IN FEASIBILITY PHASE 4 MEANS  FEAS ACHIEVED            80000410
0027                40   GO TO (50,50,50,200), NSATIS                        80000420
0028                50   CALL CONVRG (N1)                                    80000430
0029                     GO TO (60,10,125), N1                               80000440
                    C  MINIMUM ACHIEVED IF  N1=1                             80000450
0030                60   GO TO (70,80), NT3                                  80000460
0031                70   CALL TIMEC                                          80000470
0032                     CALL OUTPUT (1)                                     80000480
                    C --- NUMBER OF MINIMA ACHIEVED INCREASED BY 1           80000490
0033                80   NUMINI=NUMINI+1                                     80000500
0034                     MN=0                                                80000510
0035                     GO TO (190,90,90), NPHASE                           80000520
0036                90   CALL ESTIM                                          80000530
                    C  FINAL MIGHT HAVE BEEN CALLED BY ESTIM-CONVERGED IF N2=1  80000540
0037                     IF(NEXOP3.NE.2) GO TO 27                            80000550
0038                     CALL SENS                                           80000560
0039                  27 GO TO (100,110,120), NT4                            80000570
                    C   NT4=1 FINAL CONVERGENCE ON 0 ORDER ESTIMATES, NT4=2 CONVERGE ON FIRS80000580
                    C  ORDER ESTIMATES, NT4=3 CONVERGE ON SECOND ORDER ESTIMATES.  80000590
0040                100  CALL FINAL (NF1)                                    80000600
0041                     GO TO (130,140), NF1                                80000610
0042                110  GO TO (130,140), NF2                                80000620
0043                120  GO TO (130,140), NF3                                80000630
0044                125  NPHASE=5                                            80000640
0045                130  RETURN                                              80000650
0046                140  RHO=RHO/RATIO                                       80000660
                    C USING PREVIOSLY COMPUTED VALUES FOR F, AND RJ P IS RECOMPUTED WITH THER80000670
                    C NEW VALUE OF RHO.                                      80000680
0047                     CALL PEVALU                                         80000690
                    C A VECTOR IS LEFT IN DELX(1) BY ESTIM                   80000700
0048                     IF (NUMINI-2) 10,150,150                            80000710
0049                150  GO TO (10,160,160), NT7                             80000720
0050                160  CALL GRAD (2)                                       80000730
0051                     CALL OPT                                            80000740
0052                     GO TO (180,170), NT3                                80000750
0053                170  WRITE (6,210)                                       80000760
0054                     CALL OUTPUT (1)                                     80000770
0055                180  GO TO 50                                            80000780
0056                190  IF (G) 90,90,200                                    80000790
0057                200  RETURN                                              80000800
                    C --- DUAL VALUE GREATER THAN 0 MEANS NO FEASIBLE POINT EXISTS  80000810
                    C                                                        80000820
0058                210  FORMAT (6X,30HMOVED ON EXTRAPOLATION VECTOR )       80000830
0059                     END                                                 80000840
```

Figure 9.--Subroutine BØDY.

```
FORTRAN IV G LEVEL 21          BOUND          DATE = 80242          12/12/03

      C  SUBROUTINE BOUND WAS DEVELOPED BY GHAEMI(1979) TO CALCULATE          BOU00060
      C  UPPER AND/OR LOWER PIECEWISE LINEAR PARAMETRIC BOUNDS ON             BOU00070
      C  OPTIMAL VALUE FUNCTIONS WHEN THESE FUNCTIONS ARE KNOWN TO BE         BOU00080
      C  CONVEX OR CONCAVE OVER THE RANGE OF A GIVEN PARAMETER.               BOU00090
0001          SUBROUTINE BOUND                                              BOU00100
0002          IMPLICIT REAL*8(A-H,O-Z)                                      BOU00110
0003          COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS                        BOU00120
0004          COMMON/ABG/LY,LZ,PER(45)                                      BOU00130
0005          COMMON/ABG1/FE1,FE2                                           BOU00140
0006          COMMON/ABG2/DF1(45),DF2(45)                                   BOU00150
0007          COMMON/OP7/NEXOP7                                             BOU00160
0008          DIMENSION E(20),FQ(20),FQQ(20),FU12(20),FL12(20)              BOU00170
0009          E1=PAR(LZ)                                                    BOU00180
0010          E2=PAR(LZ) + PER(LZ)                                          BOU00190
      C LINE FIT THROUGH POINTS 1 AND 2                                     BOU00200
0011          S12 = (FE1-FE2)/(E1-E2)                                       BOU00210
0012          B12 = FE1-S12*E1                                              BOU00220
      C TANGENT LINE UNDER ESTIMATING F* AT POINT 1                         BOU00230
0013          S1=DF1(LZ)                                                    BOU00240
0014          B1=FE1-S1*E1                                                  BOU00250
      C TANGENT LINE UNDER ESTIMATING F* AT POINT 2                         BOU00260
0015          S2=DF2(LZ)                                                    BOU00270
0016          B2=FE2-S2*E2                                                  BOU00280
      C INTERSECTION OF THE ABOVE TWO LINES                                 BOU00290
0017          E3=(B2-B1)/(S1-S2)                                            BOU00300
0018          FE3=S1*(B2-B1)/(S1-S2) + B1                                   BOU00310
      C QUADRATIC FIT THROUGH POINTS 1 AND 2                                BOU00320
0019          A=(FE2-E2*S1-FE1+E1*S1)/((E1-E2)**2)                          BOU00330
0020          C=((E1**2)*(FE2-E2*S1)+(FE1-E1*S1)*(E2**2-2.*E1*E2))/((E1-E2)**2) BOU00340
0021          B=S1-2.*E1*A                                                  BOU00350
0022          AA=(FE1-E1*S2-FE2+E2*S2)/((E2-E1)**2)                         BOU00360
0023          CC=((E2**2)*(FE1-E1*S2)+(FE2-E2*S2)*(E1**2-2.*E2*E1))/((E2-E1)**2) BOU00370
0024          BB=S2-2.*E2*AA                                                BOU00380
      C BOUND EVALUATION AT 10 DISCRETE POINTS                              BOU00390
0025          E(1)=E1                                                       BOU00400
0026          DE=PER(LZ)/10.                                                BOU00410
0027          DO 580 I=1,10                                                 BOU00420
0028          IF(I.EQ.1) GO TO 750                                          BOU00430
0029          E(I)=E(I-1)+DE                                                BOU00440
      C QUADRATIC ESTIMATE                                                  BOU00450
0030        750 FQ(I)=A*E(I)**2+B*E(I)+C                                    BOU00460
0031          FQQ(I)=AA*E(I)**2+BB*E(I)+CC                                  BOU00470
      C UPPER LINEAR BOUND                                                  BOU00480
0032          FU12(I)=S12*E(I)+B12                                          BOU00490
      C LOWER LINEAR BOUND                                                  BOU00500
0033          IF(PER(LZ).GT.0.) GO TO 585                                   BOU00510
0034          IF(E(I).LT.E3) GO TO 570                                      BOU00520
0035          FL12(I)=S1*E(I)+B1                                            BOU00530
0036          GO TO 595                                                     BOU00540
0037        570 FL12(I)=S2*E(I)+B2                                          BOU00550
0038          GO TO 595                                                     BOU00560
0039        585 IF(E(I).GT.E3) GO TO 590                                    BOU00570
0040          FL12(I)=S1*E(I)+B1                                            BOU00580
0041          GO TO 595                                                     BOU00590
0042        590 FL12(I)=S2*E(I)+B2                                          BOU00600
0043        595 CONTINUE                                                    BOU00610
0044        580 CONTINUE                                                    BOU00620
      C CHOICE ON PROPER QUADRATIC FIT                                      BOU00630
0045          IF(NEXOP7 .GE.4) GO TO 810                                    BOU00640
0046          IF(FQ(5) .GT. FQQ(5)) GO TO 800                               BOU00650
0047          A=AA                                                          BOU00660
0048          B=BB                                                          BOU00670
0049          C=CC                                                          BOU00680
0050          DO 100 I=1,10                                                 BOU00690
0051          FQ(I)=FQQ(I)                                                  BOU00700
0052        100 CONTINUE                                                    BOU00710
0053          GO TO 800                                                     BOU00720
0054        810 IF(FQ(5) .LT. FQQ(5)) GO TO 800                             BOU00730
0055          A=AA                                                          BOU00740
0056          B=BB                                                          BOU00750
0057          C=CC                                                          BOU00760
0058          DO 110 I=1,10                                                 BOU00770
0059          FQ(I)=FQQ(I)                                                  BOU00780
0060        110 CONTINUE                                                    BOU00790
0061        800 CONTINUE                                                    BOU00800
0062          WRITE(6,600) LZ                                               BOU00810
0063        600 FORMAT(1H1,25X,'OPTIMAL VALUE FUNCTION BOUNDS WHEN PAR(',I2,  BOU00820
           $') IS PERTURBED',/,24X,57('*'),//)                              BOU00830
0064          WRITE(6,610) LZ,E1,LZ,FE1,LZ,E2,LZ,FE2                        BOU00840
0065        610 FORMAT(2X,'POINT 1 (UNPERTURBED SOLUTION) :',/,2X,'PAR(',I2,  BOU00850
           $')=',E15.7,10X,'F(PAR(',I2,')=',E15.7,//,2X,'POINT 2 (PERTURBED' BOU00860
           $,' SOLUTION) :',/,2X,'PAR(',I2,')=',E15.7,10X,'F(PAR(',I2,      BOU00870
           $')=',E15.7,///)                                                 BOU00880
0066          GO TO (910,920,930,940,950,960),NEXOP7                        BOU00890
      C OUTPUT WHEN NEXOP7=1                                                BOU00900
0067        910 WRITE(6,620) S12,LZ,B12                                     BOU00910
0068        620 FORMAT(2X,'LINE PASSING THROUGH POINTS 1 AND 2 AND OVER'    BOU00920
```

Figure 10.--Subroutine BØUND.

```
     $,' ESTIMATING F* ',/,1X,59('-'),//,11X,'F = ',E15.7,' * PAR(',      BOU00930
     $12,') + ',E15.7,///)                                                BOU00940
0069      WRITE(6,630) S1,LZ,B1                                           BOU00950
0070   630 FORMAT(2X,'LINE UNDER ESTIMATING F* AT POINT 1 ',/1X,36('-'),   BOU00960
     $//,11X,'F = ',E15.7,' * PAR(',I2,') + ',E15.7,///)                  BOU00970
0071      WRITE(6,640) S2,LZ,B2                                           BOU00980
0072   640 FORMAT(2X,'LINE UNDER ESTIMATING F* AT POINT 2',/,1X,36('-'),   BOU00990
     $//,11X,'F = ',E15.7,' * PAR(',I2,') + ',E15.7,///)                  BOU01000
0073      WRITE(6,650)A,LZ,B,LZ,C                                         BOU01010
0074   650 FORMAT(2X,'QUADRATIC ESTIMATION OF F* THROUGH POINTS 1 AND'     BOU01020
     $,' 2',/,1X,50('-'),  //,11X,'F = ',E15.7,' * PAR(',I2,             BOU01030
     $')**2 + ',E15.7,' * PAR(',I2,') + ', E15.7 ,///)                    BOU01040
0075      WRITE(6,660) LZ                                                 BOU01050
0076   660 FORMAT(2X,'F* BOUND EVALUATION AT TEN EQUIDISTANCE POINTS',     BOU01060
     $' BETWEEN POINTS 1 AND 2',/,1X,71('='),//,10X,'PAR(',I2,')',       BOU01070
     $10X,'LOWER BOUND',10X,'UPPER BOUND',10X,'QUAD ESTIMATE',/,          BOU01080
     $10X,7('-'),9X,13('-'),8X,13('-'),8X,15('-'),/)                      BOU01090
0077      WRITE(6,670)(E(I),FL12(I),FU12(I),FQ(I),I=1,10)                 BOU01100
0078      WRITE(6,670)E2,FE2,FE2,FE2                                      BOU01110
0079   670 FORMAT(10X,F7.3,7X,E15.7,6X,E15.7,6X,E15.7)                     BOU01120
0080      RETURN                                                          BOU01130
     C OUTPUT WHEN NEXOP7=2                                               BOU01140
0081   920 WRITE(6,620)S12,LZ,B12                                          BOU01150
0082      WRITE(6,660) LZ                                                 BOU01160
0083      WRITE(6,921)(E(I),FU12(I), I=1,10)                              BOU01170
0084      WRITE(6,921) E2,FE2                                             BOU01180
0085   921 FORMAT(10X,F7.3,28X,E15.7)                                      BOU01190
0086      RETURN                                                          BOU01200
     C OUTPUT WHEN NEXOP7=3                                               BOU01210
0087   930 WRITE(6,630) S1, LZ,B1                                          BOU01220
0088      WRITE(6,640)S2,LZ,B2                                            BOU01230
0089      WRITE(6,660)LZ                                                  BOU01240
0090      WRITE(6,931)(E(I),FL12(I),I=1,10)                               BOU01250
0091      WRITE(6,931)E2,FE2                                              BOU01260
0092   931 FORMAT(10X,F7.3,7X,E15.7)                                       BOU01270
0093      RETURN                                                          BOU01280
     C OUTPUT WHEN NEXOP7=4                                               BOU01290
0094   940 WRITE(6,941) S12,LZ,B12                                         BOU01300
0095   941 FORMAT(2X,'LINE PASSING THROUGH POINTS 1 AND 2 AND UNDER'       BOU01310
     $,' ESTIMATING F* ',/,1X,59('-'),//,11X,'F = ',E15.7,' * PAR(',      BOU01320
     $12,') + ',E15.7,///)                                                BOU01330
0096      WRITE(6,942) S1,LZ,B1                                           BOU01340
0097   942 FORMAT(2X,'LINE  OVER ESTIMATING F* AT POINT 1 ',/1X,36('-'),   BOU01350
     $//,11X,'F = ',E15.7,' * PAR(',I2,') + ',E15.7,///)                  BOU01360
0098      WRITE(6,943) S2,LZ,B2                                           BOU01370
0099   943 FORMAT(2X,'LINE  OVER ESTIMATING F* AT POINT 2',/,1X,36('-'),   BOU01380
     $//,11X,'F = ',E15.7,' * PAR(',I2,') + ',E15.7,///)                  BOU01390
0100      WRITE(6,650)A,LZ,B,LZ,C                                         BOU01400
0101      WRITE(6,660) LZ                                                 BOU01410
0102      WRITE(6,670)(E(I),FU12(I),FL12(I),FQ(I),I=1,10)                 BOU01420
0103      WRITE(6,670) E2,FE2,FE2,FE2                                     BOU01430
0104      RETURN                                                          BOU01440
     C OUTPUT WHEN NEXOP7=5                                               BOU01450
0105   950 WRITE(6,942) S1,LZ,B1                                           BOU01460
0106      WRITE(6,943) S2,LZ,B2                                           BOU01470
0107      WRITE(6,660) LZ                                                 BOU01480
0108      WRITE(6,921)(E(I),FL12(I), I=1,10)                              BOU01490
0109      WRITE(6,921) E2,FE2                                             BOU01500
0110      RETURN                                                          BOU01510
     C OUTPUT WHEN NEXOP7=6                                               BOU01520
0111   960 WRITE(6,941)S12,LZ,B12                                          BOU01530
0112      WRITE(6,660) LZ                                                 BOU01540
0113      WRITE(6,931) (E(I),FU12(I), I=1,10)                             BOU01550
0114      WRITE(6,931) E2,FE2                                             BOU01560
0115      RETURN                                                          BOU01570
0116      END                                                             BOU01580
```

Figure 10.--*continued*

DIFF2. Both results are printed out. If it is found that MATRIX creates a nonzero element below the main diagonal of A , then a switch is set to cause the statement STØP to be executed in MAIN. Subroutine CHCKER appears as Figure 11.

6.4 CØNVRG

After each iteration of the algorithm to locate the minimum of the unconstrained function (the W function) subroutine CØNVRG is called to determine if the current point is an acceptable approximation of a point giving the minimum value of the W function. The argument N2 of this routine is given a value of 1 if the point is close enough; otherwise, it is given a value of 2. CØNVRG appears as Figure 12.

6.5 DIFF1

Subroutine DIFF1 computes the first derivatives by numerical differencing. The user-supplied subroutine RESTNT is called 2n times for each gradient evaluated by DIFF1. For the function f , DIFF1 computes the *ith* component of the gradient using the formula

$$(\nabla f(x^0))_i = \frac{f(x^0 + \theta e_i) - f(x^0 - \theta e_i)}{2\theta},$$

where e_i is a vector of zeroes with a 1 in the *ith* component and θ is a small number whose value is assigned by the user. DIFF1 is shown as Figure 13.

6.6 DIFF2

Subroutine DIFF2 computes the second derivatives by numerical differencing. The matrix of second partials is computed using central differencing. The differences are calculated using the gradients of the function. The user-supplied subroutine GRAD1 is called 2n times for each matrix of second partial derivatives evaluated by DIFF2. DIFF2 is Figure 14.

```
FORTRAN IV G LEVEL  21                CHCKER            DATE = 80242        09/21/09

0001                SUBROUTINE CHCKER                                                 CHE00040
                    C                                                                 CHE00050
                    C            MARCH 1971                                           CHE00060
                    C                                                                 CHE00070
                    C CHCKER COMPUTES AND LIST THE FIRST PARTIAL DERIVATIVES USING GRAD1   CHE00080
                    C AND THEN USING NUMERICAL DIFFERENCING (DIFF1).  IF REQUESTED THE    CHE00090
                    C SECOND PARTIAL DERIVATIVES ARE COMPUTED AND LISTED USING MATRIX AND  CHE00100
                    C DIFF2.                                                          CHE00110
0002                IMPLICIT REAL*8(A-H,O-Z)                                          CHE00120
0003                REAL*4 XEP1,XEP2                                                  CHE00130
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1                CHE00140
0005                COMMON /EQAL/ M, M1, MZ                                           CHE00150
0006                COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2        CHE00160
0007                MMZ=1+M+MZ                                                        CHE00170
0008                DO 5 J=1,N                                                        CHE00180
0009                DEL(J)=1.2345678                                                  CHE00190
0010           5    CONTINUE                                                          CHE00200
0011                DO 10 I=1,MMZ                                                     CHE00210
0012                IN=I-1                                                            CHE00220
0013                WRITE (6,170) IN                                                  CHE00230
0014                CALL GRAD1 (IN)                                                   CHE00240
0015                WRITE (6,180) (DEL(J),J=1,N)                                      CHE00250
0016                CALL DIFF1 (IN)                                                   CHE00260
0017                WRITE (6,180) (DEL(J),J=1,N)                                      CHE00270
0018           10   CONTINUE                                                          CHE00280
                    C...SOMETIMES ONLY FIRST DERIVATIVES ARE TO BE CHECKED            CHE00290
0019                IF (NEXOP1.LT.4) GO TO 160                                        CHE00300
0020                WRITE (6,190)                                                     CHE00310
0021                DO 150 I=1,MMZ                                                    CHE00320
0022                IN=I-1                                                            CHE00330
0023                WRITE (6,170) IN                                                  CHE00340
0024                IT=2                                                              CHE00350
0025                DO 30 K=1,N                                                       CHE00360
0026                DO 20 J=1,N                                                       CHE00370
0027           20   A(K,J)=0.                                                         CHE00380
0028           30   CONTINUE                                                          CHE00390
0029                CALL MATRIX (IN,IT)                                               CHE00400
0030                IF (IT.EQ.1) GO TO 150                                            CHE00410
0031                DO 50 K=2,N                                                       CHE00420
0032                KM1=K-1                                                           CHE00430
0033                DO 40 J=1,KM1                                                     CHE00440
0034                IF (A(K,J).EQ.0.0) GO TO 40                                       CHE00450
0035                NEXOP1=5                                                          CHE00460
0036                WRITE (6,210) K,J                                                 CHE00470
0037                GO TO 60                                                          CHE00480
0038           40   CONTINUE                                                          CHE00490
0039           50   CONTINUE                                                          CHE00500
0040           60   DO 90 K=1,N                                                       CHE00510
0041                DO 70 J=K,N                                                       CHE00520
0042                IF (A(K,J).NE.0.0) GO TO 80                                       CHE00530
0043           70   CONTINUE                                                          CHE00540
0044                WRITE (6,220) K                                                   CHE00550
0045                GO TO 90                                                          CHE00560
0046           80   WRITE (6,200) K,(A(K,J),J=1,N)                                    CHE00570
0047           90   CONTINUE                                                          CHE00580
0048                DO 110 K=1,N                                                      CHE00590
0049                DO 100 J=1,N                                                      CHE00600
0050          100   A(K,J)=0.                                                         CHE00610
0051          110   CONTINUE                                                          CHE00620
0052                WRITE(6,115) IN                                                   CHE00630
0053          115   FORMAT (13HO CALL DIFF2(,I2,1H) )                                 CHE00640
0054                CALL DIFF2 (IN)                                                   CHE00650
0055                DO 140 K=1,N                                                      CHE00660
0056                DO 120 J=K,N                                                      CHE00670
0057                IF (A(K,J).NE.0) GO TO 130                                        CHE00680
0058          120   CONTINUE                                                          CHE00690
0059                WRITE (6,220) K                                                   CHE00700
0060                GO TO 140                                                         CHE00710
0061          130   WRITE (6,200) K,(A(K,J),J=1,N)                                    CHE00720
0062          140   CONTINUE                                                          CHE00730
0063          150   CONTINUE                                                          CHE00740
0064          160   CONTINUE                                                          CHE00750
0065                RETURN                                                            CHE00760
                    C                                                                 CHE00770
0066          170   FORMAT (   28HOCHECKER.....CONSTRAINT NO. , I3)                   CHE00780
0067          180   FORMAT (1HO,24HCHECKER.....1ST PARTIALS/(1X,E20.8,E20.8,E20.8,E20.CHE00790
                   18,E20.8,E20.8))                                                   CHE00800
0068          190   FORMAT (1HO,24HCHECKER.....2ND PARTIALS)                          CHE00810
0069          200   FORMAT (4HOROW, I3      /(1X,E20.8,E20.8,E20.8,E20.8,E20.8,E20.8)) CHE00820
0070          210   FORMAT (3H A(,I2,1H,,I2,10H) .NE. (.0)                           CHE00830
0071          220   FORMAT (4H ROW,I3,11H ALL ZEROS.)                                 CHE00840
0072                END                                                               CHE00850
```

Figure 11.--Subroutine CHCKER.

```
FORTRAN IV G LEVEL  21                CONVRG          DATE = 80242        09/21/37

0001                  SUBROUTINE CONVRG (N1)                                        CON00040
                C                                                                   CON00050
                C                   OCTOBER 1970                                    CON00060
                C                                                                   CON00070
                C AFTER EACH ITERATION OF THE ALGORITHM TO LOCATE THE MINIMUM OF THE CON00080
                C PENALTY FUNCTION, CONVRG DETERMINES IF THE CURRENT POINT IS CLOSE  CON00090
                C ENOUGH TO THE POINT GIVING THE MINIMUM VALUE OF THE P FUNCTION.   CON00100
                C     N1 SET EQUAL TO 1 IF MINIMUM HAS BEEN FOUND.                  CON00110
                C     N1 SET EQUAL TO 2 IF MINIMUM HAS NOT BEEN FOUND AND TIME IS NOT UPCON00120
                C     N1 SET EQUAL TO 3 OTHERWISE                                   CON00130
                C     DOTT SET EQUAL TO (DEL P)(INVERSE(DEL(DEL P)))(DEL P) IN OPT  CON00140
0002                  IMPLICIT REAL*8(A-H,O-Z)                                      CON00150
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2                      CON00160
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1            CON00170
0005                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10       CON00180
0006                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                         CON00190
0007                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,       CON00200
                     1RSIGI,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,           CON00210
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                   CON00220
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                  CON00230
0008                  COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2    CON00240
0009                  COMMON /TSW/ NSWW                                             CON00250
0010                  N1=2                                                          CON00260
0011                  IF (MN.LE.1) Q1=PO                                            CON00270
0012                  GO TO (10,20,30), NT9                                         CON00280
0013             10   IF(DABS(DOTT).LT.EPSI) GO TO 70                               CON00290
0014                  GO TO 40                                                      CON00300
0015             20   IF(DABS(DOTT).LT.(P1-PO)/5.0) GO TO 70                        CON00310
0016                  GO TO 40                                                      CON00320
0017             30   IF (ADELX.LT.EPSI) GO TO 70                                   CON00330
0018             40   GO TO (50,60), NSWW                                           CON00340
0019             50   IF (MN.LE.1) RETURN                                           CON00350
0020                  IF (PO+XEP2 .LT. Q1) GO TO 75                                 CON00360
0021                  WRITE (6,80)                                                  CON00370
0022                  GO TO 70                                                      CON00380
0023             60   CALL PUNCH                                                    CON00390
0024                  WRITE (6,90)                                                  CON00400
0025                  N1=3                                                          CON00410
                C                                                                   CON00420
                C     FOUND THE MINIMUM TO THE SUBPROBLEM.                          CON00430
0026                  RETURN                                                        CON00440
0027             70   N1=1                                                          CON00450
0028             75   Q1 = PO                                                       CON00460
0029                  RETURN                                                        CON00470
                C                                                                   CON00480
0030             80   FORMAT (100H APPARENTLY ROUNDOFF ERRORS PREVENT A MORE ACCURATE DECON00490
                     1TERMINATION OF THE MINIMUM OF THIS SUBPROBLEM.)               CON00500
0031             90   FORMAT (48H0**** TIME IS UP, CALLING EXIT FROM CONVRG. ****)  CON00510
0032                  END                                                           CON00520
```

Figure 12.--Subroutine CØNVRG.

```
FORTRAN IV G LEVEL  21                  DIFF1           DATE = 80242        09/22/00

0001                  SUBROUTINE DIFF1 (IN)                                      DIF00040
                C                                                                DIF00050
                C              FEBUARY 1971                                      DIF00060
                C                                                                DIF00070
                C DIFF1 COMPUTES THE FIRST DERIVATIVES BY NUMERICAL DIFFERENCING. DIF00080
                C                                                                DIF00090
                C--USER CAN CALL FOR DIFFERENCING OF SELECTED FUNCTIONS           DIF00100
0002                  IMPLICIT REAL*8(A-H,O-Z)                                   DIF00110
0003                  REAL*4 XEP1,XEP2                                           DIF00120
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1         DIF00130
0005                  COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2 DIF00140
0006                  COMMON/STIRX/XSTR(45),XSSS(45),DDLL(45)                    DIF00150
0007                  DO 10 J=1,N                                                DIF00160
0008            10    XSTR(J)=X(J)                                               DIF00170
0009                  DO 30 J=1,N                                                DIF00180
0010                  IF (J.EQ.1) GO TO 20                                       DIF00190
0011                  JM1=J-1                                                    DIF00200
0012                  X(JM1)=XSTR(JM1)                                           DIF00210
0013            20    X(J)=XSTR(J)+XEP1                                          DIF00220
0014                  CALL RESTNT (IN,ZZ2)                                       DIF00230
0015                  X(J)=XSTR(J)-XEP1                                          DIF00240
0016                  CALL RESTNT (IN,ZZ1)                                       DIF00250
0017            30    DEL(J)=(ZZ2-ZZ1)/(2.*XEP1)                                 DIF00260
0018                  X(N)=XSTR(N)                                               DIF00270
0019                  RETURN                                                     DIF00280
0020                  END                                                        DIF00290
```

Figure 13.--Subroutine DIFF1.

```
FORTRAN IV G LEVEL  21                DIFF2              DATE = 80242          09/22/21

0001                  SUBROUTINE DIFF2 (IN)                                          DIF00040
                C                                                                    DIF00050
                C              OCTOBER 1970                                          DIF00060
                C                                                                    DIF00070
                C DIFF2 COMPUTES THE SECOND DERIVATIVES BY NUMERICAL DIFFERENCING.    DIF00080
                C                                                                    DIF00090
0002                  IMPLICIT REAL*8(A-H,O-Z)                                       DIF00100
0003                  REAL*4 XEP1,XEP2                                               DIF00110
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1             DIF00120
0005                  COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2     DIF00130
0006                  COMMON/STIRX/XSTR(45),XSSS(45),DDLL(45)                        DIF00140
0007                  DO 10 J=1,N                                                    DIF00150
0008            10    XSSS(J)=X(J)                                                   DIF00160
0009                  DO 50 J=1,N                                                    DIF00170
0010                  IF (J.EQ.1) GO TO 20                                           DIF00180
0011                  JM1=J-1                                                        DIF00190
0012                  X(JM1)=XSSS(JM1)                                               DIF00200
0013            20    X(J)=XSSS(J)+XEP1                                              DIF00210
0014                  CALL GRAD1 (IN)                                                DIF00220
0015                  DO 30 I=1,N                                                    DIF00230
0016            30    DDLL(I)=DEL(I)                                                 DIF00240
0017                  X(J)=XSSS(J)-XEP1                                              DIF00250
0018                  CALL GRAD1 (IN)                                                DIF00260
0019                  DO 40 I=J,N                                                    DIF00270
0020            40    A(J,I)=(DDLL(I)-DEL(I))/(2.*XEP1)                              DIF00280
0021            50    CONTINUE                                                       DIF00290
0022                  X(N)=XSSS(N)                                                   DIF00300
0023                  RETURN                                                         DIF00310
0024                  END                                                            DIF00320
```

Figure 14.--Subroutine DIFF2.

6.7 ESTIM

After the minimum of the W function for a given value of r has been located, ESTIM is called. When the minimum of the W function for a given value of r is determined a subproblem is said to be solved. ESTIM performs the computations to estimate the Lagrange multipliers and make the first- and second-order estimates of the final solution of the problem. ESTIM is Figure 15.

6.8 EVALU

In the normal phase, EVALU calls the user-supplied routines to evaluate the objective function and the constraint functions at the current point x . As the constraint functions are being computed, the code performs the calculations to compute the penalty terms of the penalty function. In the feasibility phase this routine puts the negative sum of the violated constraints, which is the objective function of the entry problem, in location F . In location F the current value of the objective function is stored for use by the program. After F has been computed, the value of the penalty function is computed. Also subroutine EVALU computes a value for the dual objective function. This value is only correct at the solution to a subproblem and the value is stored in the location labeled G . EVALU appears as Figure 16.

6.9 FEAS

Subroutine FEAS determines whether the starting point is an interior feasible point or not. If the variables of a problem are supposed to be nonnegative and some of the components of the starting point vector are nonpositive, then those components are changed to small positive numbers. A modification of FEAS must be made if it is desired to change this number. If the starting point does not satisfy the nontrivial constraints (the constraints other than the nonnegativity constraints), then the program goes into the feasibility phase. In this phase the negative of the sum of all the violated inequality constraints

```
FORTRAN IV G LEVEL  21                ESTIM              DATE = 80242          09/22/45

0001                  SUBROUTINE ESTIM                                                  EST00040
                C                                                                       EST00050
                C              OCTOBER 1970                                             EST00060
                C                                                                       EST00070
                C ESTIM PERFORMS THE COMPUTATIONS TO ESTIMATE THE LAGRANGE MULTIPLIERS   EST00080
                C AND MAKE THE FIRST- AND SECOND-ORDER ESTIMATES OF THE FINAL SOLUTION   EST00090
                C OF THE PROBLEM.                                                       EST00100
0002                  IMPLICIT REAL*8(A-H,O-Z)                                          EST00110
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                    EST00120
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1                EST00130
0005                  COMMON /EQAL/ M, M1, MZ                                           EST00140
0006                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10           EST00150
0007                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                             EST00160
0008                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,           EST00170
                     1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,               EST00180
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                       EST00190
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                      EST00200
0009                  COMMON /CONPAR/ NF1,NF2,NF3                                       EST00210
0010                  CALL STORE                                                        EST00220
0011                  Z10=RATIO**2                                                      EST00230
0012                  Z9=RATIO                                                          EST00240
0013                  Z1=1.0/Z9+1.0/Z10                                                 EST00250
0014                  Z2=Z1+1./Z9**3                                                    EST00260
0015                  Z3=1./Z9**3                                                       EST00270
0016                  Z4=Z10+Z9                                                         EST00280
0017                  Z5=Z9**3                                                          EST00290
0018                  Z6=1.0/((Z10-1.0)*(Z9-1.0))                                       EST00300
0019                  Z7=1./Z9                                                          EST00310
0020                  Z8=1./(Z9-1.)                                                     EST00320
0021                  RQ=1.0/RHO                                                        EST00330
0022                  IF (NUMINI-2) 150,80,10                                           EST00340
0023            10    WRITE (6,330)                                                     EST00350
0024                  PO=(PR2-Z4*PR1+Z5*P1)*Z6                                          EST00360
0025                  G=(RATIO*G1-GR1)/(RATIO-1.)                                       EST00370
0026                  DO 20 I=1,N                                                       EST00380
0027            20    X(I)=(XR2(I)-Z4*XR1(I)+Z5*X1(I))*Z6                               EST00390
0028                  NP=NPHASE                                                         EST00400
0029                  NPHASE=4                                                          EST00410
0030                  CALL EVALU                                                        EST00420
0031                  NPHASE=NP                                                         EST00430
0032                  CALL OUTPUT (2)                                                   EST00440
                C CHECK TO SEE IF ESTIMATES HAVE CONVERGED                              EST00450
0033                  GO TO (70,30,70), NPHASE                                          EST00460
0034            30    DO 50 J=1,M                                                       EST00470
0035                  IF (RJ(J)) 40,50,50                                               EST00480
0036            40    IF (THETAO+RJ(J)) 70,50,50                                        EST00490
0037            50    CONTINUE                                                          EST00500
0038                  GO TO (70,70,60), NT4                                             EST00510
0039            60    CALL FINAL (NF3)                                                  EST00520
0040            70    CONTINUE                                                          EST00530
0041            80    WRITE (6,340)                                                     EST00540
0042                  PO=(Z9*P1-PR1)*Z8                                                 EST00550
0043                  G=(RATIO*G1-GR1)/(RATIO-1.)                                       EST00560
0044                  DO 90 I=1,N                                                       EST00570
0045            90    X(I)=(Z9*X1(I)-XR1(I))*Z8                                         EST00580
0046                  NP=NPHASE                                                         EST00590
0047                  NPHASE=4                                                          EST00600
0048                  CALL EVALU                                                        EST00610
0049                  NPHASE=NP                                                         EST00620
0050                  CALL OUTPUT (2)                                                   EST00630
                C CHECK TO SEE IF ESTIMATES HAVE CONVERGED                              EST00640
0051                  GO TO (140,100,140), NPHASE                                       EST00650
0052            100   DO 120 J=1,M                                                      EST00660
0053                  IF (RJ(J)) 110,120,120                                            EST00670
0054            110   IF (RJ(J)+THETAO) 140,120,120                                     EST00680
0055            120   CONTINUE                                                          EST00690
0056                  GO TO (140,130,140), NT4                                          EST00700
0057            130   CALL FINAL (NF2)                                                  EST00710
0058            140   CONTINUE                                                          EST00720
0059            150   WRITE (6,350)                                                     EST00730
0060                  IF (M) 180,180,160                                                EST00740
0061            160   DO 170 J=1,M                                                      EST00750
0062            170   RJ(J)=RHO/RJ1(J)                                                  EST00760
0063            180   IF (MZ) 210,210,190                                               EST00770
0064            190   DO 200 J=1,MZ                                                     EST00780
0065                  MNJ=M+J                                                           EST00790
0066            200   RJ(MNJ)=2.*RJ1(MNJ)*RQ                                           EST00800
0067            210   GO TO (220,240), NT2                                              EST00810
0068            220   DO 230 I=1,N                                                      EST00820
0069            230   X(I)=RHO/X1(I)                                                    EST00830
0070            240   CALL OUTPUT (2)                                                   EST00840
0071                  CALL REJECT                                                       EST00850
0072                  IF (NUMINI-2) 280,300,250                                         EST00860
0073            250   GO TO (280,310,260), NT7                                          EST00870
                C     SECOND  ORDER MOVE  FOR NEXT MINIMUM                            EST00880
0074            260   DO 270 I=1,N                                                      EST00890
0075            270   DELX(I)=Z1*X1(I)-Z2*XR1(I)+Z3*XR2(I)                              EST00900
```

Figure 15.--Subroutine ESTIM.

```
0076        280   PR2=PR1                                                    EST00910
0077              GR2=GR1                                                    EST00920
0078              PR1=P1                                                     EST00930
0079              GR1=G1                                                     EST00940
0080              DO 290 I=1,N                                               EST00950
0081              XR2(I)=XR1(I)                                              EST00960
0082        290   XR1(I)=X1(I)                                               EST00970
0083              RETURN                                                     EST00980
0084        300   GO TO (280,310,310), NT7                                   EST00990
0085        310   DO 320 I=1,N                                               EST01000
0086        320   DELX(I)=(X1(I)-XR1(I))*Z7                                  EST01010
0087              GO TO 280                                                  EST01020
                  C                                                          EST01030
0088        330   FORMAT (/26HO    2ND  ORDER ESTIMATES )                    EST01040
0089        340   FORMAT (/26HO    1ST  ORDER ESTIMATES )                    EST01050
0090        350   FORMAT (/25HO  LAGRANGE MULTIPLIERS  )                     EST01060
0091              END                                                        EST01070
```

Figure 15.--*continued*

```
FORTRAN IV G LEVEL 21                EVALU              DATE = 80242        09/28/17

0001                  SUBROUTINE EVALU                                                EVA00040
                C                                                                     EVA00050
                C              OCTOBER 1970                                           EVA00060
                C                                                                     EVA00070
                C IN THE NORMAL PHASE EVALU CALLS THE USER-SUPPLIED ROUTINES TO EVALUATEEVA00080
                C THE OBJECTIVE FUNCTION AND THE CONSTRAINT FUNCTIONS AT THE CURRENT   EVA00090
                C POINT. IN THE FEASIBILITY PHASE THIS ROUTINE PUTS THE NEGATIVE SUM OF EVA00100
                C THE VIOCATED CONSTRAINTS IN LOCATION F.                             EVA00110
0002                  IMPLICIT REAL*8(A-H,O-Z)                                        EVA00120
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                  EVA00130
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NMI              EVA00140
0005                  COMMON /EQAL/ H, H1, MZ                                         EVA00150
0006                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10         EVA00160
0007                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                           EVA00170
0008                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         EVA00180
                     1RSIG1,G1,X1(45),X2(45),X3(45),R2(45),XR1(45),PR1,               EVA00190
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      EVA00200
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                     EVA00210
0009                  H=0.0                                                           EVA00220
0010                  RSIGMA=0.0                                                      EVA00230
0011                  F=0.0                                                           EVA00240
0012                  NSATIS=2                                                        EVA00250
0013                  GO TO (10,100,190,200), NPHASE                                  EVA00260
                C     =1  FEASIBILITY                                                 EVA00270
                C     =2  NORMAL                                                      EVA00280
                C     =3  GUESS                                                       EVA00290
                C     =4  ALL FUNCTIONS ARE TO BE EVALUATED                           EVA00300
                C FEASIBILITY                                                         EVA00310
0014            10    GO TO (20,40), NT2                                              EVA00320
                C NON-NEGATIVIES INCLUDED                                             EVA00330
0015            20    DO 30 I=1,N                                                     EVA00340
0016                  IF (X(I)) 260,260,30                                            EVA00350
0017            30    RSIGMA=RSIGMA-RHO*DLOG(X(I))                                    EVA00360
0018            40    IF (M.EQ.0) GO TO 90                                            EVA00370
0019                  DO 80 J=1,M                                                     EVA00380
0020                  CALL RESTNT (J,RJ(J))                                           EVA00390
0021                  IF (RJ(J).LE.0.0) GO TO 50                                      EVA00400
0022                  IF (RJ(J).GT.0.0) GO TO 60                                      EVA00410
                C VIOLATION OF A PREVIOUSLY SATISFIED CONSTRAINT                      EVA00420
0023                  GO TO 260                                                       EVA00430
0024            50    IF (RJ(J).GT.0.0) GO TO 70                                      EVA00440
                C ALL VIOLATED CONSTRAINTS ADDED INTO OBJECTIVE FUNCTION              EVA00450
0025                  F=F-RJ(J)                                                       EVA00460
0026                  GO TO 80                                                        EVA00470
0027            60    RSIGMA=RSIGMA-RHO*DLOG(RJ(J))                                   EVA00480
0028                  GO TO 80                                                        EVA00490
                C INDICATES SATISFACTION OF CONSTRAINT(1ORMORE)                       EVA00500
0029            70    NSATIS=1                                                        EVA00510
0030                  RSIGMA=RSIGMA-RHO*DLOG(RJ(J))                                   EVA00520
0031            80    CONTINUE                                                        EVA00530
0032            90    CONTINUE                                                        EVA00540
                C EQUALITIES NOT COMPUTED IN FEAS. PHASE                              EVA00550
0033                  PO=F+RSIGMA                                                     EVA00560
0034                  G=F-RHO*DFLOAT(M)                                               EVA00570
0035                  IF(NT2.EQ.1) G=G-RHO*DFLOAT(N)                                  EVA00580
0036                  RETURN                                                          EVA00590
                C  REGULAR  PHASE                                                     EVA00600
0037            100   GO TO (110,130), NT2                                            EVA00610
                C NON NEGATIVITIES INCLUDED                                           EVA00620
0038            110   DO 120 I=1,N                                                    EVA00630
0039                  IF (X(I)) 260,260,120                                           EVA00640
0040            120   RSIGMA=RSIGMA-RHO*DLOG(X(I))                                    EVA00650
0041            130   IF (M.EQ.0) GO TO 150                                           EVA00660
0042                  DO 140 J=1,M                                                    EVA00670
0043                  CALL RESTNT (J,RJ(J))                                           EVA00680
0044                  IF (RJ(J).LE.0.0) GO TO 260                                     EVA00690
0045                  RSIGMA=RSIGMA-RHO*DLOG(RJ(J))                                   EVA00700
0046            140   CONTINUE                                                        EVA00710
                C EVALUATE AND ADD IN EQUALITY CONSTRAINTS                            EVA00720
0047            150   CONTINUE                                                        EVA00730
0048                  CALL RESTNT (0,F)                                               EVA00740
0049                  IF (MZ) 180,180,160                                             EVA00750
0050            160   DO 170 I=1,MZ                                                   EVA00760
0051                  J=I+M                                                           EVA00770
0052                  CALL RESTNT (J,RJ(J))                                           EVA00780
                C ADD INTO THIRD TERM OF P FUNCTION                                   EVA00790
0053                  H=H+(RJ(J))**2                                                  EVA00800
0054            170   CONTINUE                                                        EVA00810
0055                  H=H/RHO                                                         EVA00820
0056            180   PO=RSIGMA+H                                                     EVA00830
0057                  PO=F+PO                                                         EVA00840
0058                  G=2.*H-RHO*DFLOAT(M)                                            EVA00850
0059                  G=G+F                                                           EVA00860
0060                  IF(NT2.EQ.1) G=G-RHO*DFLOAT(NI                                  EVA00870
                C DUAL VALUE                                                          EVA00880
0061                  RETURN                                                          EVA00890
                C GUESS PHASE NOT CODED                                               EVA00900
```

Figure 16.-- Subroutine EVALU.

```
0062        190   RETURN                                                    EVA00910
            C---  STRAIGHT FUNCTION EVALUATION  ( MAJN+FEAS ONLY)          EVA00920
0063        200   CONTINUE                                                  EVA00930
0064              IF (M.EQ.0) GO TO 220                                     EVA00940
0065              DO 210 I=1,M                                              EVA00950
0066              CALL RESTNT (I,RJ(I))                                     EVA00960
0067        210   CONTINUE                                                  EVA00970
0068        220   CALL RESTNT (0,F)                                         EVA00980
            C EQUALITY CONSTRAINTS                                          EVA00990
0069              IF (MZ) 250,250,230                                       EVA01000
0070        230   DO 240 I=1,MZ                                             EVA01010
0071              KZ=M+I                                                    EVA01020
0072        240   CALL RESTNT (KZ,RJ(KZ))                                   EVA01030
0073        250   RETURN                                                    EVA01040
            C CONSTRAINTS VIOLATED NOT SO BEFORE                            EVA01050
0074        260   NSATIS=3                                                  EVA01060
0075              P0=10.E35                                                 EVA01070
0076              RETURN                                                    EVA01080
0077              END                                                       EVA01090
```

Figure 16.--*continued*

is labeled for use as the objective function in the entry problem. Then the routine calls BØDY to minimize this auxiliary function subject to the set of satisfied constraints. In the feasibility phase if a violated constraint is fortuitously satisfied an immediate return to FEAS is made and a new entry problem is begun, including the newly satisfied constraint in the set of constraints active for the entry problem. Thus the entry problem can result in a series of NLP problems being partly solved.

FEAS also contains tests that indicate when a problem does not contain a feasible starting point. Such information is printed out, and control is returned to MAIN with indicators set so that MAIN begins to try the next NLP problem in a stack of problems. When the entry problem is not a convex programming problem the test indicates only that the program is in a region from which it will be unable to locate a feasible starting point. The user, by supplying another starting point, may cuase the algorithm to generate another sequence of points that does lead to a feasible starting point. Figure 17 shows FEAS.

6.10 FINAL

Subroutine FINAL contains the test used to determine if a point satisfies the final convergence criterion of the algorithm. If a point does satisfy the criterion chosen by use of option 5 then N2, the argument of this routine, is given a value of 1; otherwise it is given a value of 2. FINAL is called following the computation of the solution estimates, which are made after the solution of the subproblem. FINAL appears as Figure 18.

6.11 GRAD

The gradient of the W function is given by the formula

$$\nabla W(x,r) = \nabla f(x) - \sum_{i=1}^{m} \frac{r}{g_i(x)} \nabla g_i(x) \sum_{i=m+1}^{m+p} \frac{2h_i(x)}{r} \nabla h_i(x) .$$

```
FORTRAN IV G LEVEL  21                FEAS                DATE = 80242        09/23/33

0001                  SUBROUTINE FEAS                                                  FEA00040
                C                                                                      FEA00050
                C              AUGUST 1971                                             FEA00060
                C                                                                      FEA00070
                C FEAS DETERMINES WHETHER THE STARTING POINT IS FEASABLE. IF IT IS NOT, FEA00080
                C FEAS LOOKS FOR A FEASABLE ONE. IF NONE EXISTS, A MESSAGE IS PRINTED  FEA00090
                C AND CONTROL RETURNS TO MAIN.                                         FEA00100
0002                  IMPLICIT REAL*8(A-H,O-Z)                                         FEA00110
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                   FEA00120
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1               FEA00130
0005                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10          FEA00140
0006                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                            FEA00150
0007                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,          FEA00160
                     1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,               FEA00170
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                       FEA00180
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                      FEA00190
0008                   NPHASE=1                                                        FEA00200
0009                   GO TO (10,50), NT2                                              FEA00210
0010             10    NFIX=1                                                          FEA00220
0011                   DO 30 I=1,N                                                     FEA00230
0012                   IF (X(I)) 20,20,30                                              FEA00240
0013             20    NFIX=2                                                          FEA00250
0014                   X(I)=1.E-05                                                     FEA00260
0015             30    CONTINUE                                                        FEA00270
0016                   GO TO (50,40), NFIX                                             FEA00280
0017             40    NPHASE=4                                                        FEA00290
0018                   CALL EVALU                                                      FEA00300
                C JUST GET ALL CONSTRAINTS EVALUATED                                   FEA00310
0019                   NPHASE=1                                                        FEA00320
0020                   WRITE (6,130)                                                   FEA00330
0021                   CALL OUTPUT (2)                                                 FEA00340
0022             50    IF (M) 90,90,60                                                 FEA00350
0023             60    DO 70 I=1,M                                                     FEA00360
0024                   IF (RJ(I)) 100,100,70                                           FEA00370
0025             70    CONTINUE                                                        FEA00380
0026             80    CALL TIMEC                                                      FEA00390
0027                   WRITE (6,140)                                                   FEA00400
0028                   G=0.0                                                           FEA00410
0029                   CALL RESTNT (0,F)                                               FEA00420
0030                   CALL OUTPUT (2)                                                 FEA00430
0031             90    RETURN                                                          FEA00440
0032             100   CALL BODY                                                       FEA00450
0033                   IF(NPHASE .EQ. 5) RETURN                                        FEA00460
0034                   DO 110 I=1,M                                                    FEA00470
0035                   IF (RJ(I)) 120,120,110                                          FEA00480
0036             110   CONTINUE                                                        FEA00490
0037                   GO TO 80                                                        FEA00500
0038             120   WRITE (6,150)                                                   FEA00510
                C TO INDICATE TO MAIN TO START ON NEXT PROBLEM.                        FEA00520
0039                   NPHASE=5                                                        FEA00530
0040                   GO TO 90                                                        FEA00540
                C                                                                      FEA00550
0041             130   FORMAT (1HO,2X,48HMADE VIOLATED NON-NEGATIVITIES SLIGHTLY POSITIVEFEA00560
                     1)                                                                FEA00570
0042             140   FORMAT (51HO*****THE FEASIBLE STARTING POINT TO BE USED IS ...)  FEA00580
0043             150   FORMAT (3X,89HTHIS PROBLEM POSSESSES NO FEASIBLE STARTING POINT, WFEA00590
                     1I'LL LOOK FOR DATA FOR NEXT PROBLEM.  )                          FEA00600
0044                   END                                                             FEA00610
```

Figure 17.--Subroutine FEAS.

```
FORTRAN IV G LEVEL 21          FINAL          DATE = 80242          09/28/56

                                                                              FIN00040
0001            SUBROUTINE FINAL (N2)                                         FIN00050
                C                                                             FIN00060
                C            OCTOBER 1970                                     FIN00070
                C                                                             FIN00080
                C FINAL CONTAINS THE TESTS USED TO DETERMINE WHETHER A POINT SATISFIES  FIN00090
                C THE FINAL CONVERGENCE CRITERION CHOOSEN TO DETERMINE IF THE NLP       FIN00100
                C PROBLEM HAS BEEN SOLVED.                                    FIN00110
                C N2 SET EQUAL TO 1 IF CONVERGENCE CRITERION IS SATISFIED.    FIN00120
                C N2 SET EQUAL TO 2 OTHERWISE.                                FIN00130
0002                 IMPLICIT REAL*8(A-H,O-Z)                                 FIN00140
0003                 REAL*4 RHOIN,RATIO,EPSI,THETAO                           FIN00150
0004                 COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1       FIN00160
0005                 COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                    FIN00170
0006                 COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,  FIN00180
                    1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,       FIN00190
                    2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),               FIN00200
                    3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS              FIN00210
0007                 GO TO (10,20,30), NT5                                    FIN00220
0008            10   EPSIL=DABS(F/G-1.)                                       FIN00230
0009                 IF (EPSIL-THETAO) 50,50,70                               FIN00240
0010            20   IF(DABS(RSIGMA)-THETAO) 50,50,70                         FIN00250
0011            30   IF (NUMINI-1) 50,40,40                                   FIN00260
0012            40   PEST=PR1-(PR1-PO)/(1.-1./SQRT(RATIO))                    FIN00270
0013                 EPSIL=DABS(PEST/G-1.)                                    FIN00280
0014                 IF (EPSIL-THETAO) 50,70,70                               FIN00290
0015            50   N2=1                                                     FIN00300
0016                 GO TO (80,60), NT6                                       FIN00310
0017            60   CALL PUNCH                                               FIN00320
0018                 GO TO 80                                                 FIN00330
0019            70   N2=2                                                     FIN00340
0020            80   RETURN                                                   FIN00350
0021                 END                                                      FIN00360
```

Figure 18.--Subroutine FINAL.

Subroutine GRAD calls the user-supplied subroutine GRAD1 to compute $\nabla f(x)$, $\nabla g_i(x)$, and $\nabla h_i(x)$ and performs the computations to evaluate $\nabla W(x,r)$. The negative of the gradient of the W function (i.e., $-\nabla W$) is left in the array DELXO when GRAD returns control to the calling routine. When the argument of GRAD has a value of 2 this is all that is done. However, when it has a value of 1, GRAD also does part of the computations needed to evaluate the matrix of second partial derivatives of W at x . Subroutine GRAD is shown in Figure 19.

6.12 INVERS

When Newton's method is used to minimize the W function for a given value of r , it maps the negative of the gradient of W with the inverse of the matrix of second partial derivatives of W evaluated at x . A positive definite matrix will always give a vector along which the value of W will initially decrease. That is, for iteration i , a move is made along the vector S^i , given by the formula

$$S^i = -[\nabla^2 W(x,r)]^{-1} \nabla W(x,r) .$$

This is equivalent to solving the set of simultaneous linear equations

$$[\nabla^2 W(x,r)]S^i = -\nabla W(x,r)$$

for S^{i*}, where $\nabla^2 W$ and ∇W are respectively the matrix of second derivatives and gradient vector of W evaluated at (x,r) .

Subroutine INVERS solves this set of equations for S^i using an L-U decomposition method (the Crout procedure). If it is determined that the matrix of second partials is not positive definite, a different procedure is used to obtain a direction S^i . A complete discussion of this procedure is given on page 167 of Fiacco and McCormick [10]. When

---

*In the program the vector S^i is called DELX.

FORTRAN IV G LEVEL 21 GRAD DATE = 80242 12/07/35

```
0001            SUBROUTINE GRAD (IS)                                                GRA00040
                C                                                                   GRA00050
                C            OCTOBER 1970                                           GRA00060
                C                                                                   GRA00070
                C GRAD COMPUTES THE GRADIENT OF THE PENALTY FUNCTION AND THE OUTER  GRA00080
                C PRODUCT FACTORS OF THE MATRIX OF SECOND PARTIALS OF P.             GRA00090
                C  IF  (IS=1) ACCUM. MATRIX OF 2ND PARTIALS  IF(IS=2) DONT          GRA00100
0002                IMPLICIT REAL*8(A-H,O-Z)                                        GRA00110
0003                REAL*4 RHOIN,RATIO,EPSI,THETAO                                  GRA00120
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              GRA00130
0005                COMMON /EQAL/ M, M1, MZ                                         GRA00140
0006                COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10         GRA00150
0007                COMMON/VALUE/F,G,P0,RSIGMA,RJ(90),RHO                           GRA00160
0008                COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         GRA00170
                   1RSIG1,G1,XL(45),X2(45),X3(45),XR2(45),XR1(45),PR1,              GRA00180
                   2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      GRA00190
                   3 PREV3,ADELX, NCTR, NUMINI, NPHASE, NSATIS                      GRA00200
0009                GO TO (10,30), IS                                               GRA00210
0010            10  DO 20 I=1,N                                                     GRA00220
0011                DO 20 J=1,I                                                     GRA00230
0012            20  A(I,J)=0.                                                       GRA00240
0013            30  DO 40 I=1,N                                                     GRA00250
0014            40  DELXO(I)=0.                                                     GRA00260
                C THIS SECTION WORKS CORRECTLY IN FEASIBILITY PHASE AS WELL AS NORMAL PHGRA00270
0015                GO TO (50,80), NT2                                              GRA00280
0016            50  DO 70 I=1,N                                                     GRA00290
0017                DELXO(I)=-RHO/X(I)                                              GRA00300
0018                GO TO (60,70), IS                                               GRA00310
0019            60  A(I,I)=(-DELXO(I)/X(I))                                         GRA00320
0020            70  CONTINUE                                                        GRA00330
0021            80  CONTINUE                                                        GRA00340
0022                IF (M.LE.0) GO TO 180                                           GRA00350
0023                DO 170 K=1,M                                                    GRA00360
0024                CALL GRAD1 (K)                                                  GRA00370
0025                IF (RJ(K).GT.0.0) GO TO 110                                     GRA00380
                C ALL VIOLATED CONSTRAINT  GRADS ADDED TO OBJ. FUNCTION             GRA00390
0026                DO 100 I=1,N                                                    GRA00400
0027                IF (DEL(I)) 90,100,90                                           GRA00410
0028            90  DELXO(I)=DELXO(I)-DEL(I)                                       GRA00420
0029            100 CONTINUE                                                        GRA00430
0030                GO TO 170                                                       GRA00440
0031            110 TT=RHO/RJ(K)                                                    GRA00450
0032                DO 160 I=1,N                                                    GRA00460
0033                IF (DEL(I)) 120,160,120                                         GRA00470
                C IF DEL(I)=0 SKIP ALL THE FOLLOWING COMPUTATION INVOLVING * BY DEL(I) GRA00480
0034            120 T=TT*DEL(I)                                                     GRA00490
0035                DELXO(I)=DELXO(I)-T                                             GRA00500
0036                GO TO (130,160), IS                                             GRA00510
0037            130 T=T/RJ(K)                                                       GRA00520
0038                DO 150 JJ=1,I                                                   GRA00530
0039                IF (DEL(JJ)) 140,150,140                                        GRA00540
0040            140 A(I,JJ)=A(I,JJ)+T*DEL(JJ)                                       GRA00550
0041            150 CONTINUE                                                        GRA00560
0042            160 CONTINUE                                                        GRA00570
0043            170 CONTINUE                                                        GRA00580
                C  EQUALITY CHANGES FOR GRAD                                        GRA00590
0044            180 IF (MZ.LE.0) GO TO 250                                          GRA00600
0045                GO TO (250,190,250), NPHASE                                     GRA00610
0046            190 RQ=2./RHO                                                       GRA00620
0047                DO 240 J=1,MZ                                                   GRA00630
0048                K=M+J                                                           GRA00640
0049                CALL GRAD1 (K)                                                  GRA00650
0050                TT=RQ*RJ(K)                                                     GRA00660
0051                DO 230 I=1,N                                                    GRA00670
0052                IF (DEL(I).EQ.0.0) GO TO 230                                    GRA00680
0053                DELXO(I)=DELXO(I)+DEL(I)*TT                                     GRA00690
0054                GO TO (200,230), IS                                             GRA00700
0055            200 T=RQ*DEL(I)                                                     GRA00710
0056                DO 220 JJ=1,I                                                   GRA00720
0057                IF (DEL(JJ)) 210,220,210                                        GRA00730
0058            210 A(I,JJ)=A(I,JJ)+T*DEL(JJ)                                       GRA00740
0059            220 CONTINUE                                                        GRA00750
0060            230 CONTINUE                                                        GRA00760
0061            240 CONTINUE                                                        GRA00770
0062            250 GO TO (260,280), IS                                             GRA00780
0063            260 DO 270 I=1,N                                                    GRA00790
0064            270 DIAG(I)=A(I,I)                                                  GRA00800
0065            280 GO TO (290,330,290), NPHASE                                     GRA00810
                C LEAVES NEGATIVE GRADIENT IN DELP                                  GRA00820
0066            290 DO 300 I=1,N                                                    GRA00830
0067            300 DELXO(I)=-DELXO(I)                                              GRA00840
0068            310 ADELX=0.                                                        GRA00850
0069                DO 320 I=1,N                                                    GRA00860
0070            320 ADELX=ADELX+DELXO(I)**2                                         GRA00870
0071                ADELX=DSQRT(ADELX)                                              GRA00880
0072                RETURN                                                          GRA00890
0073            330 CALL GRAD1 (0)                                                  GRA00900
0074                DO 340 I=1,N                                                    GRA00910
0075            340 DELXO(I)=-DELXO(I)-DEL(I)                                       GRA00920
                C LEAVES THE NEG. GRAD OF P IN DELXO                                GRA00930
0076                GO TO 310                                                       GRA00940
0077                END                                                             GRA00950
```

Figure 19.--Subroutine GRAD.

this occurs the program prints out "ORTHOGONAL MOVE." This also warns the user that the problem is probably not a convex program. Subroutine INVERS appears as Figure 20.

6.13 LMULT

Subroutine LMULT, called by subroutine SENS, calculates the Lagrange multiplier sensitivities according to steps 5 and 6 of Algorithm 2.1 (Section 2). LMULT is given in Figure 21.

6.14 MAIN

MAIN is the program that initiates the SENSUMT algorithm to solve a nonlinear programming (NLP) problem; it is not a subroutine. The input of parameters and options, as well as a starting point, is done in MAIN, and the call to READIN (a user-supplied subroutine) is made to allow the user to read in data needed to evaluate his objective function and constraint functions. Starting point data or blank cards should always be supplied by the user because starting point data cards are always read in MAIN. Subroutine FEAS is called to obtain a feasible starting point by making use of the SENSUMT algorithm to solve the entry problem if the user-supplied point is not feasible. The actual solution of the NLP problem using the SENSUMT algorithm is supervised by subroutine BØDY. The calls to SENS (sensitivity subroutine) and BØUND (bound subroutine) are also done in MAIN. When calculating bounds, after the solution and sensitivity analysis of the unperturbed problem, MAIN reiterates SENSUMT to solve the subject problem with the perturbed data. Perturbations and readjustments in the problem data are done in PERT (perturbation subroutine), on call by MAIN. Subroutine MAIN is shown as Figure 22.

6.15 ØPT

The purpose of subroutine ØPT is to obtain a $\bar{\theta} > 0$ such that $W(\bar{x} + \bar{\theta}\bar{S})$ is a minimum with respect to θ along the vector $\bar{x} + \theta\bar{S}$.

```
FORTRAN IV G LEVEL 21                INVERS          DATE = 80242        12/20/20

0001            SUBROUTINE INVERS (NSME)                                        INV00040
                C                                                               INV00050
                C            OCTOBER 1970                                       INV00060
                C                                                               INV00070
                C INVERS SOLVES THE SET OF EQUATIONS FOR THE MOVE-VECTOR USING THE   INV00080
                C CROUT PROCEDURE. IF THE MATRIX IS NOT POSITIVE DEFINITE, A DIFFERENT  INV00090
                C METHOD IS USED.                                               INV00100
                C                                                               INV00110
                C****PERFORMING A L-U DECOMPOSITION OF THE MATRIX A, TAKING ADVANAGE OF  INV00120
                C****THE SYMMETRY OF THE A MATRIX.                              INV00130
                C****IF A NON-POSITIVE PIVOT CANDIDATE IS GENERATED, THEN MCCORMICK,S   INV00140
                C****PROCEDURE IS USED(SEE PP. 167-168 IN FIACCO AND MCCORMICK).  INV00150
                C                                                               INV00160
                C****IF NSME =1 WORKING WITH A NEW A MATRIX.  IF NSME= 2 USING PREVIOUS  INV00170
                C****A MATRIX, BUT HAVE A NEW RIGHT-HAND-SIDE.                  INV00180
                C                                                               INV00190
                C****NINV IS THE NUMBER OF NON-POSITIVE PIVOT CANDIDATES GENERATED.  INV00200
0002                 IMPLICIT REAL*8(A-H,O-Z)                                   INV00210
0003                 REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2                   INV00220
0004                 COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1         INV00230
0005                 COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10    INV00240
0006                 COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,    INV00250
                    1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,         INV00260
                    2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                 INV00270
                    3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                INV00280
0007                 COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2 INV00290
0008                  DIMENSION B(45)                                           INV00300
0009                 EQUIVALENCE (B,DELX)                                       INV00310
0010                 GO TO(20, 170), NSME                                       INV00320
0011            20   NINV=0                                                     INV00330
0012                 IF (A(1,1)) 40,30,50                                       INV00340
0013            30   NINV=1                                                     INV00350
0014                 GO TO 70                                                   INV00360
0015            40   NINV=1                                                     INV00370
0016            50   A(1,1)=1./A(1,1)                                           INV00380
0017                 DO 60 I=2,N                                                INV00390
0018            60   A(1,I)=A(1,I)*A(1,1)                                       INV00400
0019            70   DO 160 J=2,N                                               INV00410
0020                 JM1=J-1                                                    INV00420
0021                 T=0.                                                       INV00430
0022                 DO 90 I=1,JM1                                              INV00440
0023                 IF (A(I,J)) 80,90,80                                       INV00450
0024            80   T=T+A(J,I)*A(I,J)                                          INV00460
0025            90   CONTINUE                                                   INV00470
0026                 A(J,J)=A(J,J)-T                                            INV00480
0027                 IF (A(J,J)) 110,100,120                                    INV00490
0028            100  NINV=NINV+1                                                INV00500
0029                 GO TO 170                                                  INV00510
0030            110  NINV=NINV+1                                                INV00520
0031            120  A(J,J)=1./A(J,J)                                           INV00530
0032                 IF (J.EQ.N) GO TO 170                                      INV00540
0033                 JP1=J+1                                                    INV00550
0034                 DO 150 L=JP1,N                                             INV00560
0035                 T=0.                                                       INV00570
0036                 DO 140 I=1,JM1                                             INV00580
0037                 IF (A(I,J)) 130,140,130                                    INV00590
0038            130  T=T+A(L,I)*A(I,J)                                          INV00600
0039            140  CONTINUE                                                   INV00610
0040                 A(L,J)=A(L,J)-T                                            INV00620
0041                 A(J,L)=A(L,J)*A(J,J)                                       INV00630
0042            150  CONTINUE                                                   INV00640
0043            160  CONTINUE                                                   INV00650
0044            170  CONTINUE                                                   INV00660
0045                 IF (NINV) 180,180,290                                      INV00670
0046            180  B(1)=B(1)*A(1,1)                                           INV00680
0047                 DO 210 J=2,N                                               INV00690
0048                 T=0.                                                       INV00700
0049                 JM1=J-1                                                    INV00710
0050                 DO 200 I=1,JM1                                             INV00720
0051                 IF (A(J,I)) 190,200,190                                    INV00730
0052            190  T=T+A(J,I)*B(I)                                            INV00740
0053            200  CONTINUE                                                   INV00750
0054                 B(J)=(B(J)-T)*A(J,J)                                       INV00760
0055            210  CONTINUE                                                   INV00770
0056                 DO 240 I=1,NM1                                             INV00780
0057                 NMK=N-I                                                    INV00790
0058                 DO 230 J=1,I                                               INV00800
0059                 L=NP1-J                                                    INV00810
0060                 IF (A(NMK,L)) 220,230,220                                  INV00820
0061            220  B(NMK)=B(NMK)-A(NMK,L)*B(L)                                INV00830
0062            230  CONTINUE                                                   INV00840
0063            240  CONTINUE                                                   INV00850
0064            250  GO TO (280,260), NT3                                       INV00860
0065            260  WRITE (6,430)                                              INV00870
0066                 WRITE (6,420) (DELXO(I),I=1,N)                             INV00880
0067            270  WRITE (6,440)                                              INV00890
0068                 WRITE (6,420) (DELX(I),I=1,N)                              INV00900
```

Figure 20.--Subroutine INVERS.

```
0069        280   RETURN                                                  INV00910
            C---  COMPUTE ORTHOGONAL MOVE                                INV00920
0070        290   CONTINUE                                                INV00930
0071              DO 350 II=1,N                                           INV00940
0072              I=N-II+1                                                INV00950
0073              IF (A(I,I)) 310,300,320                                 INV00960
0074        300   B(I)=0.0                                                INV00970
0075              GO TO 350                                               INV00980
0076        310   B(I)=1.0                                                INV00990
0077              GO TO 330                                               INV01000
0078        320   B(I)=0.0                                                INV01010
0079        330   IP1=I+1                                                 INV01020
0080              IF (IP1.GT.N) GO TO 350                                 INV01030
0081              DO 340 J=IP1,N                                          INV01040
0082        340   B(I)=B(I)-A(I,J)*B(J)                                   INV01050
0083        350   CONTINUE                                                INV01060
0084              GO TO 360                                               INV01070
            C-- CHECK MAYBE DO DIFF FOR P.S.D.                            INV01080
0085        360   ZC2=0.0                                                 INV01090
0086              DO 370 I=1,N                                            INV01100
0087        370   ZC2=ZC2+DELX0(I)*B(I)                                   INV01110
0088              IF (ZC2) 380,400,400                                    INV01120
0089        380   DO 390 I=1,N                                            INV01130
0090        390   B(I)=-B(I)                                              INV01140
0091        400   WRITE (6,450)                                           INV01150
            C MCC ZANGWILL  ONE MOD                                       INV01160
0092              IF (NEXOP2.NE.2) GO TO 250                              INV01170
0093              DO 410 K=1,N                                            INV01180
0094        410   B(K)=B(K)+DELX0(K)                                      INV01190
0095              GO TO 250                                               INV01200
            C                                                             INV01210
0096        420   FORMAT (7E17.8)                                         INV01220
0097        430   FORMAT (1HO,6X,12HDEL P VECTOR)                         INV01230
0098        440    FORMAT(1HO,6X,24HSECOND ORDER MOVE VECTOR)             INV01240
0099        450   FORMAT (1HO,6X,15HORTHOGONAL MOVE)                      INV01250
0100              END                                                     INV01260
```

Figure 20.--*continued*

```
FORTRAN IV G LEVEL  21                LMULT                DATE = 80242          12/08/23

                                                                                              LMU00040
0001                  SUBROUTINE LMULT(IND,DELMU,DEM,DU)                                      LMU00050
                C                                                                             LMU00060
                C                        1 MARCH 1976                                         LMU00070
                C                                                                             LMU00080
                C  SUBROUTINE LMULT IS USED TO ESTIMATE THE PARTIAL DERIVATIVES OF THE        LMU00090
                C  LAGRANGE MULTIPLIERS.  IT IS CALLED BY SENS WHEN NEXOP4 = 1.  USING       LMU00100
                C  THE DIFFERENTIABILITY OF X(I), IT TAKES ADVANTAGE OF THE RELATIONS         LMU00110
                C  U(I) = RHO/G(I)**2 AND W(J) = 2*H(J)/RHO, DIFFERENTIATING THEM WITH       LMU00120
                C  RESPECT TO THE PARAMETERS. THIS SUBROUTINE WAS CODED BY R.L. ARMACOST      LMU00130
0002                  IMPLICIT REAL*8(A-H,O-Z)                                                LMU00140
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2                                LMU00150
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1                      LMU00160
0005                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                                   LMU00170
0006                  COMMON/EQAL/ M, M1, M2                                                  LMU00180
0007                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,                 LMU00190
                     1NTCTR,NUMIN1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,                  LMU00200
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                              LMU00210
                     3PREV3,ADELX,RSIG1,G1,NPHASE,NSATIS                                      LMU00220
0008                  DIMENSION DELMU(90),DU(90),KTEST(45)                                    LMU00230
0009                  MMZ = M + MZ                                                            LMU00240
0010                  GO TO (1,2,3,3), IND                                                    LMU00250
                C IND = 0                                                                     LMU00260
0011                  DO 100 I=1,MMZ                                                          LMU00270
0012            100   DELMU(I) = 0.                                                           LMU00280
0013                  RETURN                                                                  LMU00290
                C IND = 1                                                                     LMU00300
0014            1     DO 50 I=1,MMZ                                                           LMU00310
0015            50    DELMU(I) = RJ(I)                                                        LMU00320
0016                  RETURN                                                                  LMU00330
                C IND = 2                                                                     LMU00340
0017            2     DO 60 I=1,MMZ                                                           LMU00350
0018            60    DELMU(I) = (DELMU(I) - RJ(I))/DEM                                       LMU00360
0019                  RETURN                                                                  LMU00370
0020            3     DO 70 I=1,MMZ                                                           LMU00380
0021                  CALL GRADI(I)                                                           LMU00390
0022                  CALL RESTNT(I,VAL)                                                      LMU00400
0023                  SUM = 0.                                                                LMU00410
0024                  DO 71 JJ=1,N                                                            LMU00420
0025            71    SUM = SUM + DEL(JJ)*DELX(JJ)                                            LMU00430
0026                  IF(IND.EQ.4) GO TO 80                                                   LMU00440
0027                  DU(I) =-(SUM + DELMU(I))*RHO/(VAL**2)                                   LMU00450
0028                  GO TO 70                                                                LMU00460
0029            80    DU(I) = 2.*(SUM + DELMU(I))/RHO                                         LMU00470
0030            70    CONTINUE                                                                LMU00480
0031                  RETURN                                                                  LMU00490
0032                  END                                                                     LMU00500
```

Figure 21.--Subroutine LMULT.

```
FORTRAN IV G LEVEL 21                MAIN              DATE = 80242        12/12/30

                C                                                                       MAI00060
                C                AUGUST 1971                                            MAI00070
                C                                                                       MAI00080
                C MAIN IS THE PROGRAM THAT INITIATES THE SUMT ALGORITHM. THE INPUT OF   MAI00090
                C PARAMETERS, OPTIONS, AND STARTING POINT IS DONE IN MAIN. AFTER THE    MAI00100
                C SOLUTION OF ONE NLP PROBLEM MAIN LOOKS FOR DATA FOR ANOTHER NLP PROB. MAI00110
                C  MAIN HAS BEEN MODIFIED BY AKMACOST(1976) TO INCLUDE SENSITIVITY      MAI00120
                C  ROUTINES AND BY GHAEMI(1979) TO INCLUDE BOUND CALCULATIONS           MAI00130
0001                    IMPLICIT REAL*8(A-H,O-Z)                                        MAI00140
0002                    REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2,TMMAX                  MAI00150
0003                    COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              MAI00160
0004                    COMMON /EQAL/ M, M1, MZ                                         MAI00170
0005                    COMMON /OPTNS/ NT1,NT2,NT3,NT4,N.5,NT6,NT7,NT8,NT9,NT10         MAI00180
0006                    COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                           MAI00190
0007                    COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         MAI00200
                       1RSIGL,GL,X1(45),X2(45),X3(45),XP2(45),XR1(45),PR1,              MAI00210
                       2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      MAI00220
                       3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                     MAI00230
0008                    COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS                          MAI00240
0009                    COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2      MAI00250
0010                    COMMON/ABG/LY,LZ,PER(45)                                        MAI00260
0011                    COMMON/ABG1/FE1,FE2                                             MAI00270
0012                    COMMON/ABG2/DF1(45),DF2(45)                                     MAI00280
0013                    COMMON/OP6/NEXOP6                                               MAI00290
0014                    COMMON/OP7/NEXOP7                                               MAI00300
0015                    COMMON/ABG4/XY(45)                                              MAI00310
0016                    COMMON/ABG6/XX(45)                                              MAI00320
0017                    DIMENSION XZ(45)                                                MAI00330
                C       PARAMETER CARD                                                  MAI00340
0018            10      READ(5,50,END=40) EPSI,RHOIN,THETAO,RATIO,TMMAX,M,N,MZ          MAI00350
0019                    LY=0.                                                           MAI00360
0020                    LZ=0.                                                           MAI00370
0021                    RRR=RHOIN                                                       MAI00380
0022            20      CONTINUE                                                        MAI00390
0023                    IF(LY .EQ. 0.) GO TO 15                                         MAI00400
0024                    IF(NEXOP3 .EQ. 0.) GO TO 40                                     MAI00410
0025               15 CALL SET (TMMAX)                                                  MAI00420
                C       INITIAL  X VERCTOR CARD  FORMAT                                 MAI00430
0026                    IF(LY .GT. 0.) GO TO 200                                        MAI00440
0027                    READ (5,60) (X(I),I=1,N)                                        MAI00450
0028                    DO 4000 I=1,N                                                   MAI00460
0029                    XZ(I)=X(I)                                                      MAI00470
0030             4000 CONTINUE                                                          MAI00480
0031              200 RHOIN=RRR                                                         MAI00490
0032                    NTCTR=0                                                         MAI00500
0033                    NP1=N+1                                                         MAI00510
0034                    NM1=N-1                                                         MAI00520
                C SUBROUTINE READIN IS UNDER PROGRAMMER CONTROL                         MAI00530
0035                    IF(LY .GT. 0.) GO TO 210                                        MAI00540
0036                    CALL READIN                                                     MAI00550
                C OPTION CARD FOLLOWS PROGRAMMERS DATA                                  MAI00560
0037                    READ (5,80) NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10           MAI00570
0038                    GO TO 215                                                       MAI00580
0039              210 CALL READIN                                                       MAI00590
0040                    CALL PERT                                                       MAI00600
0041                    IF(LZ .GT. NPAR) GO TO 40                                       MAI00610
0042              215 WRITE (6,110)                                                     MAI00620
0043                    WRITE (6,120) N,M,MZ,TMMAX,RHOIN,RATIO,EPSI,THETAO              MAI00630
0044                    WRITE (6,130)                                                   MAI00640
0045                    WRITE (6,80) NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10           MAI00650
                C--READ TOLERANCES                                                      MAI00660
0046                    IF(LY .GT. 0.) GO TO 220                                        MAI00670
0047                    READ (5,60) XEP1, XEP2                                          MAI00680
0048              220 WRITE (6,90)                                                      MAI00690
0049                    WRITE (6,70) XEP1, XEP2                                         MAI00700
                C--READ SECOND OPTION CARD                                              MAI00710
0050                    IF(LY .GT. 0.) GO TO 230                                        MAI00720
0051                    READ (5,80) NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,NEXOP6, NEXOP7  MAI00730
0052              230 WRITE (6,100)                                                     MAI00740
0053                    WRITE (6,80) NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,NEXOP6,NEXOP7  MAI00750
0054                    CALL TIMEC                                                      MAI00760
0055                    NPHASE=4                                                        MAI00770
                C --- JUST TO GET AN INITIAL PRINTOUT                                   MAI00780
0056                    CALL EVALU                                                      MAI00790
0057                    PO=0.0                                                          MAI00800
0058                    G=0.0                                                           MAI00810
0059                    H=0.0                                                           MAI00820
0060                    RSIGMA=0.0                                                      MAI00830
0061                    CALL OUTPUT (2)                                                 MAI00840
0062                    CALL STORE                                                      MAI00850
0063                    IF (NEXOP1.GT.1) CALL CHCKER                                    MAI00860
0064                    IF (NEXOP1.EQ.3) STOP 01072                                     MAI00870
0065                    IF (NEXOP1.EQ.5) STOP 01104                                     MAI00880
0066                    CALL FEAS                                                       MAI00890
                C NPHASE=5 INDICATES EITHER TIME RAN OUT OR NO FEASIBLE POINT WAS FOUND MAI00900
0067                    GO TO (30,30,30,30,10), NPHASE                                  MAI00910
0068            30      NPHASE=2                                                        MAI00920
```

Figure 22.--MAIN

```
0069              NTCTR=0                                                    MAI00930
0070              CALL BUDY                                                  MAI00940
0071              IF(NPHASE.EQ.5) GO TO 17                                   MAI00950
0072              IF(NEXOP3.EQ.1) GO TO 18                                   MAI00960
0073              IF(NEXOP3.EQ.3) GO TO 19                                   MAI00970
                C FOLLOWING STTATEMENT IS INCLUDED FOR WATER POLLUTION PROJECT ONLY  MAI00980
                C IT HAS TO BE REMOVED LATER (NT8=5 IMPLIES POLLUTION PROJECT)      MAI00990
0074              IF(NT8 .EQ. 5) CALL PROF                                   MAI01000
0075              GO TO 17                                                   MAI01010
0076           18 ISENS = 4                                                  MAI01020
0077              CALL SENS                                                  MAI01030
0078              IF(LY.GT.0.) CALL BOUND                                    MAI01040
0079              LY=LY+1                                                    MAI01050
0080              GO TO 17                                                   MAI01060
0081           19 ISENS = 2                                                  MAI01070
0082              DO 21 I=1,12                                               MAI01080
0083              CALL SENS                                                  MAI01090
0084              IF(LY.GT.0.) CALL BOUND                                    MAI01100
0085              LY=LY+1                                                    MAI01110
0086           21 CONTINUE                                                   MAI01120
0087           17 DO 300 I=1,N                                               MAI01130
0088              X(I)=XZ(I)                                                 MAI01140
0089          300 CONTINUE                                                   MAI01150
0090              GO TO 20                                                   MAI01160
0091        40    STOP                                                       MAI01170
                C                                                            MAI01180
                C     PARAMETER CARD                                         MAI01190
0092        50    FORMAT (5E12.0,3I4)                                        MAI01200
                C     INITIAL  X VERCTOR CARD  FORMAT                        MAI01210
0093        60    FORMAT (6E12.6)                                            MAI01220
0094        70    FORMAT (6E20.7)                                            MAI01230
                C     OPTION CARD FORMAT                                     MAI01240
                C     OPTION CARD FORMAT                                     MAI01250
0095        80    FORMAT (10I7)                                              MAI01260
0096        90    FORMAT(13H0 TOLERANCES  )                                  MAI01270
0097        100   FORMAT (26H0 SECOND SET OF OPTIONS     )                   MAI01280
0098        110   FORMAT (56H1 NONLINEAR PROGRAMMING ROUTINE-SUMT VERSION 4 03/01/73MAI01290
                  1       )                                                  MAI01300
0099        120   FORMAT (1H0,5X,2HN=I3,6X,2HM=I3,6X,3HMZ=I3//8X,10HMAX. TIME=E14.7,MAI01310
                  14X,2HR=E14.7,4X,6HRATIO=E14.7,6X,8HEPSILON=E14.7,4X,6HTHETA=E14.7)MAI01320
0100        130   FORMAT (18H0 OPTIONS SELECTED)                             MAI01330
0101              END                                                        MAI01340
```

Figure 22.--*continued*

AD-A096 693 GEORGE WASHINGTON UNIV WASHINGTON DC INST FOR MANAGE--ETC F/G 9/2
A USER'S MANUAL FOR SENSUMT: A PENALTY FUNCTION COMPUTER PROGRA--ETC(U)
OCT 80 A V FIACCO, A GHAEMI DAAG29-79-C-0062
UNCLASSIFIED SERIAL-T-434 ARO-16229.9-M NL

2 OF 2

END
DATE FILMED
4-81
DTIC

The vector $\bar{S}$, along which the search for the minimum is made, is determined in subroutine XMOVE. The method used to locate the minimum along the vector is the Golden Section search method, which is based on the Fibonacci search method (see [10, p. 193]).

The Golden Section method is a one-dimensional search method to find the minimum of the function that does not require the computation of derivatives. Figure 23 shows ØPT.

6.16 ØUTPUT

Subroutine ØUTPUT contains most of the "write" statements used to print out information on the results of solving an NLP problem. It is used to print out information after each iteration and also to print out the solution estimates and the estimates of the Lagrange multipliers. ØUTPUT is given in Figure 24.

6.17 PARDIF

The subroutine PARDIF is called by subroutine SENS to assign a differencing increment for use in the central differencing formulas indicated in Steps 2 and 4 of Algorithm 2.1. It assigns a fixed value to the differencing interval when the sensitivity analysis is conducted at the final subproblem, or assigns a value over a specified interval for sensitivity analysis at the final subproblem. When a trajectory sensitivity analysis is performed, PARDIF returns a differencing interval which is dependent on the particular subproblem involved, refining the value as the subproblem approaches the final one. Subroutine PARDIF is shown in Figure 25.

6.18 PERT

Subroutine PERT, called by the program MAIN, performs the bookkeeping regarding various parametric changes and adjustments required in the process of bound calculation. Figure 26 is subroutine PERT.

```
FORTRAN IV G LEVEL  21                OPT                DATE = 80243          10/18/39

0001                  SUBROUTINE OPT                                                    OPT00040
                C                                                                       OPT00050
                C                MARCH 1971                                             OPT00060
                C                                                                       OPT00070
                C OPT LOOKS FOR A MINIMUM ALONG THE SEARCH VECTOR USING THE GOLDEN      OPT00080
                C SECTION SEARCH METHOD.                                                OPT00090
0002                  IMPLICIT REAL*8(A-H,O-Z)                                          OPT00100
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                    OPT00110
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1                OPT00120
0005                  COMMON/VALUE/F,G,P0,RSIGMA,RJ(90),RHO                             OPT00130
0006                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,           OPT00140
                     1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,               OPT00150
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                       OPT00160
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                      OPT00170
0007                  KSW=1                                                             OPT00180
0008                  N405=1                                                            OPT00190
0009                  P31=P0                                                            OPT00200
0010                  ISW=1                                                             OPT00210
0011                  DOTT=0.                                                           OPT00220
0012                  DO 10 J=1,N                                                       OPT00230
0013             10   DOTT=DOTT+DELX(J)*DELXO(J)                                        OPT00240
0014                  GO TO 40                                                          OPT00250
0015             20   DO 30 I=1,N                                                       OPT00260
0016             30   DELX(I)=-DELX(I)                                                  OPT00270
0017             40   CONTINUE                                                          OPT00280
0018                  N404=0                                                            OPT00290
0019                  MN=MN+1                                                           OPT00300
                C MN IS NOW NUMB. OF POINTS AFTER MIN ACHIEVED                          OPT00310
0020                  NTCTR=NTCTR+1                                                     OPT00320
0021                  DO 50 I=1,N                                                       OPT00330
0022             50   X2(I)=X(I)                                                        OPT00340
0023                  PX1=P0                                                            OPT00350
0024                  N401=0                                                            OPT00360
0025             60   N401=N401+1                                                       OPT00370
0026                  DO 70 I=1,N                                                       OPT00380
0027             70   X(I)=X2(I)+DELX(I)                                                OPT00390
0028                  CALL EVALU                                                        OPT00400
                C 1 MEANS SATIS.OF CONSTRAINT NT.PREV. 2MEANS NOCHANGE 3MEANS VIOLATION OPT00410
                C IF POINT IS NOT FEASIBLE GIVE IT AN ARBRITRARILY HIGH VALUE           OPT00420
0029                  GO TO (540,90,80), NSATIS                                         OPT00430
0030             80   PX2=10.E35                                                        OPT00440
0031                  P0=10.E35                                                         OPT00450
0032                  GO TO 100                                                         OPT00460
0033             90   CONTINUE                                                          OPT00470
0034                  PX2=P0                                                            OPT00480
0035                  IF (PX1-PX2) 100,100,150                                          OPT00490
0036            100   IF (N401-2) 130,110,110                                           OPT00500
0037            110   DO 120 I=1,N                                                      OPT00510
0038            120   X1(I)=X(I)                                                        OPT00520
0039                  P1=PX2                                                            OPT00530
0040                  GO TO 430                                                         OPT00540
                C ONLY ONE  POINT  SO FAR COMPUTED                                      OPT00550
0041            130   DO 140 I=1,N                                                      OPT00560
0042            140   X3(I)=X2(I)                                                       OPT00570
0043                  PREV3=PX1                                                         OPT00580
0044                  GO TO 180                                                         OPT00590
0045            150   DO 160 I=1,N                                                      OPT00600
0046                  X3(I)=X2(I)                                                       OPT00610
0047                  X2(I)=X(I)                                                        OPT00620
0048            160   DELX(I)=1.61803399*DELX(I)                                        OPT00630
0049                  PREV3=PX1                                                         OPT00640
0050                  PX1=PX2                                                           OPT00650
0051                  GO TO 60                                                          OPT00660
                C GOLDEN SECTION SEARCH METHOD.                                         OPT00670
                C B VECTOR  GOES TO X1(I)                                               OPT00680
0052            170   P0=1.E36                                                          OPT00690
0053                  N404=N404+1                                                       OPT00700
0054            180   DO 190 I=1,N                                                      OPT00710
0055            190   X1(I)=X(I)                                                        OPT00720
0056                  P1=P0                                                             OPT00730
0057                  DO 200 I=1,N                                                      OPT00740
0058                  X(I)=.38196601*(X1(I)-X3(I))+X3(I)                                OPT00750
0059            200   X2(I)=X(I)                                                        OPT00760
0060                  CALL EVALU                                                        OPT00770
0061                  GO TO (540,270,210), NSATIS                                       OPT00780
0062            210   IF (N404.LT.30) GO TO 170                                         OPT00790
0063            211   CONTINUE                                                          OPT00800
                C THERE IS NO REFERENCE TO 211, THE ABOVE STATEMENT IS A DUMMY STATEMENTOPT00810
                C--IT IS POSSIBLE NO FEASIBLE POINT EXIST, IF NOT TRY MOVING ON DELXO.  OPT00820
                C--  IF IT IS NOT POSSIBLE TO MOVE ON DELXO THEN WE MUST BE AT A        OPT00830
                C--  SOLUTION OF NLP PROBLEM.                                           OPT00840
0064                  IF (N404.GT.100) GO TO 240                                        OPT00850
0065            220   DO 230 I=1,N                                                      OPT00860
0066                  IF(DABS(DABS(X3(I)/X1(I))-1.).GT.1.E-7) GO TO 170                 OPT00870
0067            230   CONTINUE                                                          OPT00880
0068            240   GO TO (250,260), N405                                             OPT00890
```

Figure 23.--Subroutine ØPT.

```
0069        250   N405=2                                                    OPT00900
                  C--TRY TO MOVE ON GRADIENT.                               OPT00910
0070              NTCTR=NTCTR-1                                             OPT00920
0071              MM=MM-1                                                   OPT00930
0072              GO TO 20                                                  OPT00940
0073        260   WRITE (6,580)                                             OPT00950
0074              CALL TIMEC                                                OPT00960
0075              CALL OUTPUT (1)                                           OPT00970
0076              CALL REJECT                                               OPT00980
0077              STOP 22042                                                OPT00990
                  C                                                         OPT01000
0078        270   CONTINUE                                                  OPT01010
0079              N404=0                                                    OPT01020
0080              PX1=P0                                                    OPT01030
0081              DO 280 I=1,N                                              OPT01040
0082        280   X(I)=0.38196601*(X1(I)-X2(I))+X2(I)                       OPT01050
0083              CALL EVALU                                                OPT01060
0084              GO TO (540,290,220), NSATIS                               OPT01070
0085        290   PX2=P0                                                    OPT01080
0086              N401=1                                                    OPT01090
0087        300   N401=N401+1                                               OPT01100
0088              IF (N401-25) 340,310,310                                  OPT01110
0089        310   KSW=2                                                     OPT01120
0090              IF (N401-40) 320,460,460                                  OPT01130
0091        320   DO 330 I=1,N                                              OPT01140
0092              IF(DABS(X2(I)/X(I)-1.0).GE.1.E-7) GO TO 340               OPT01150
0093        330   CONTINUE                                                  OPT01160
0094              GO TO 460                                                 OPT01170
0095        340   IF(DABS(PX1/PX2-1.).LE.1.E-7) GO TO 460                   OPT01180
0096              IF (PX1-PX2) 350,460,400                                  OPT01190
                  C FROM LEFTORIGHT  X3(I)(PREV3)X2(I)(PX1)X(I)PX2 X1(I)P1  OPT01200
0097        350   DO 360 I=1,N                                              OPT01210
0098        360   X1(I)=X(I)                                                OPT01220
                  C THROW AWAY RIGHT PART                                   OPT01230
0099              P1=PX2                                                    OPT01240
0100              DO 370 I=1,N                                              OPT01250
                  C POINTXP1 BECOMES XP2                                    OPT01260
0101        370   X(I)=.38196601*(X1(I)-X3(I))+X3(I)                        OPT01270
                  C TEMPORARILY IN  X STORAGE                               OPT01280
0102              CALL EVALU                                                OPT01290
0103              GO TO (540,380,170), NSATIS                               OPT01300
0104        380   CONTINUE                                                  OPT01310
0105              PX2=PX1                                                   OPT01320
                  C SWITCH VECTORS TO PROPER POSITION                       OPT01330
0106              PX1=P0                                                    OPT01340
0107              DO 390 I=1,N                                              OPT01350
0108              XX=X2(I)                                                  OPT01360
0109              X2(I)=X(I)                                                OPT01370
0110        390   X(I)=XX                                                   OPT01380
0111              GO TO 300                                                 OPT01390
                  C LEFT SIDE  TOSSED  AWAY                                 OPT01400
                  C-- CHANGES FOR NONUNIMODAL FN                            OPT01410
                  C-- GO TO THROW AWAY RIGHT IN THIS CASE(INIT VAL LT FIB PT OPT01420
0112        400   IF (PREV3-PX2) 350,350,410                                OPT01430
0113        410   DO 420 I=1,N                                              OPT01440
0114              X3(I)=X2(I)                                               OPT01450
0115        420   X2(I)=X(I)                                                OPT01460
0116              PREV3=PX1                                                 OPT01470
0117              PX1=PX2                                                   OPT01480
0118        430   DO 440 I=1,N                                              OPT01490
0119        440   X(I)=0.38196601*(X1(I)-X2(I))+X2(I)                       OPT01500
0120              CALL EVALU                                                OPT01510
0121              GO TO (540,450,170), NSATIS                               OPT01520
0122        450   CONTINUE                                                  OPT01530
0123              PX2=P0                                                    OPT01540
0124              GO TO 300                                                 OPT01550
                  C THE INTERIOR POINTS NOW GIVE EQUAL VALUE FOR P. COMPUTE MIDPOINT.  OPT01560
0125        460   DO 470 I=1,N                                              OPT01570
0126              DELX0(I)=X(I)                                             OPT01580
0127              X(I)=(DELX0(I)+X2(I))*0.5                                 OPT01590
0128        470   CONTINUE                                                  OPT01600
0129              CALL EVALU                                                OPT01610
0130              GO TO (480,490), KSW                                      OPT01620
0131        480   IF(DABS(P0/PX1-1.).GT.1.E-7) GO TO 520                    OPT01630
0132        490   GO TO (500,510), ISW                                      OPT01640
0133        500   IF (P0.LT.P31) GO TO 510                                  OPT01650
0134              ISW=2                                                     OPT01660
                  C IF P-FUNCTION DION,T  GO DOWN TRY NEG VECT.             OPT01670
0135              GO TO 20                                                  OPT01680
0136        510   RETURN                                                    OPT01690
0137        520   DO 530 I=1,N                                              OPT01700
0138        530   X(I)=DELX0(I)                                             OPT01710
0139              GO TO 350                                                 OPT01720
                  C ARE  WE NOW IN FEASIBILITY PHASE                        OPT01730
0140        540   DO 550 I=1,M                                              OPT01740
0141              IF (RJ(I)) 560,560,550                                    OPT01750
0142        550   CONTINUE                                                  OPT01760
0143              NSATIS=4                                                  OPT01770
0144              RETURN                                                    OPT01780
                  C---  PROBLEM HAS BECOME FEASIBLE                         OPT01790
                  C ---  P - FUNCTION CHANGES IF A CONSTRAINT BECOMES FEASIBLE  OPT01800
0145        560   MM=0                                                      OPT01810
0146              DO 570 I=1,M                                              OPT01820
0147        570   RJ(I)=RJ(I)                                               OPT01830
0148              RETURN                                                    OPT01840
                  C                                                         OPT01850
0149        580   FORMAT ( 80H  OPT CANoT FIND A FEASIBLE POINT,THAT GIVES A LOWER VOPT01860
                  1ALUE OF THE P-FUNCTION.          )                       OPT01870
0150              END                                                       OPT01880
```

Figure 23.-- *continued*

```
FORTRAN IV G LEVEL  21                OUTPUT              DATE = 80242        12/12/50

0001                  SUBROUTINE OUTPUT (K)                                          OUT00060
                C                                                                    OUT00070
                C              OCTOBER 1970                                          OUT00080
                C                                                                    OUT00090
                C OUTPUT PRINTS OUT INFORMATION ON THE RESULTS OF EACH ITERATION AND THEOUT00100
                C SOLUTION ESTIMATES AND THE ESTIMATES OF THE LAGRANGE MULTIPLIERS    OUT00110
                C THIS ROUTINE WAS MODIFIED BY GHAEMI(1979) WHILE INCORPORATING      OUT00120
                C BOUND CALCULATION TECHNIQUE.                                       OUT00130
0002                  IMPLICIT REAL*8(A-H,O-Z)                                       OUT00140
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                 OUT00150
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1             OUT00160
0005                  COMMON /EQAL/ M, M1, MZ                                        OUT00170
0006                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10        OUT00180
0007                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                          OUT00190
0008                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,        OUT00200
                     1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,             OUT00210
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                     OUT00220
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                    OUT00230
0009                  COMMON/ABG/LY,LZ,PER(45)                                       OUT00240
0010                  COMMON/ABG1/FE1,FE2                                            OUT00250
0011                  COMMON/ABG4/XY(45)                                             OUT00260
0012                  COMMON/ABG2/DF1(45),DF2(45)                                    OUT00270
0013                  COMMON/OP6/NEXOP6                                              OUT00280
0014                  COMMON/ABG5/TX(45),TDELX(45)                                   OUT00290
0015                  COMMON/ABG6/XX(45)                                             OUT00300
0016                  NZ=M+MZ                                                        OUT00310
0017                  GO TO (10,20), K                                               OUT00320
0018            10    WRITE (6,60)                                                   OUT00330
0019                  WRITE (6,70) NTCTR,DOTT,RHO,ADELX,NPHASE                       OUT00340
0020            20    WRITE (6,80) F,PO,G,RSIGMA,M                                   OUT00350
0021                  IF(LY .GT. 0.) GO TO 200                                       OUT00360
0022                  DO 210 I=1,N                                                   OUT00370
0023                  XY(I)=X(I)                                                     OUT00380
0024              210 CONTINUE                                                       OUT00390
0025                  FE1=F                                                          OUT00400
0026                  GO TO 220                                                      OUT00410
0027            200   FE2=F                                                        OUT00420
0028            220   WRITE (6,90) (X(I),I=1,N)                                    OUT00430
                C CALCULATION OF GP VARIABLES                                      OUT00440
0029                  IF(NEXOP6.NE.1) GO TO 315                                      OUT00450
0030                  INDEX=0.                                                       OUT00460
0031                  CALL TRANS(INDEX)                                              OUT00470
0032                  WRITE(6,300)(X(I),I=1,N)                                       OUT00480
0033                  DO 410 I=1,N                                                   OUT00490
0034              410 X(I)=TX(I)                                                     OUT00500
0035              315 WRITE (6,110)                                                  OUT00510
0036                  GO TO (30,40), NT2                                             OUT00520
0037            30    WRITE (6,120)                                                  OUT00530
0038                  WRITE (6,100) (RJ(I),I=1,NZ)                                   OUT00540
0039                  GO TO 50                                                       OUT00550
0040            40    WRITE (6,100) (RJ(I),I=1,NZ)                                   OUT00560
0041              300 FORMAT(/,6X,'CORRESPONDING VALUE OF GP VARIABLE IS',/,(6E20.7)) OUT00570
0042               50 RETURN                                                       OUT00580
                C                                                                  OUT00590
0043            60    FORMAT (50H0**********                          )            OUT00600
0044            70    FORMAT (10X,6HPOINT=I4,6X,6H DOTT=E15.7,6X,4HRHO=E15.7,6X,10HMAGNIOUT00610
                     1TUDE=E15.7,6X,6HPHASE=I2)                                    OUT00620
0045            80    FORMAT (8X,2HF=E15.7,5X,2HP=E15.7,5X,2HG=E15.7,5X,7HRSIGMA=E15.7,5OUT00630
                     1X,2HM=E15.7)                                                 OUT00640
0046            90    FORMAT (6X,25HTHE CURRENT VALUE OF X IS/(6E20.7))            OUT00650
0047            100   FORMAT (6E20.7)                                              OUT00660
0048            110   FORMAT (6X,21HTHE CONSTRAINT VALUES)                         OUT00670
0049            120   FORMAT (28X,34HNOT INCLUDING THE NON-NEGATIVITIES)           OUT00680
0050                  END                                                          OUT00690
```

Figure 24.--Subroutine ØUTPUT.

```
FORTRAN IV G LEVEL  21                PARDIF              DATE = 80242          12/09/35

0001                  SUBROUTINE PARDIF                                          PAR00040
                C                                                                PAR00050
                C                     15 MARCH 1972                              PAR00060
                C                                                                PAR00070
                C  THIS SUBROUTINE RETURNS A DIFFERENCING INTERVAL FOR DPAR(20) WHEN   PAR00080
                C  CALLED BY SUBROUTINE SENS.  ISENS CONTROLS THE VALUE OF THE INTERVAL PAR00090
                C  WITH A MINIMUM VALUE IN THE NEIGHBORHOOD OF 1.E-15 DEPENDING ON XEP1.PAR00100
                C  ISENS DEPENDS ON THE KIND OF SENSITIVITY ANALYSIS BEING CONDUCTED.   PAR00110
                C  THE SUBROUTINE WAS CODED BY R. L. ARMACOST.                   PAR00120
                C                                                                PAR00130
0002                  IMPLICIT REAL*8(A-H,O-Z)                                   PAR00140
0003                  REAL*4 XEP1                                                PAR00150
0004                  COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2 PAR00160
0005                  COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS                     PAR00170
0006                  MULT = MIN0(ISENS,8)                                       PAR00180
0007                  XXEP1 = XEP1 / (10.**MULT)                                 PAR00190
0008                  IF(XXEP1.LE.0.) XXEP1 = 1.E-10                             PAR00200
0009                  DO 10 I=1,NPAR                                             PAR00210
0010                10 DPAR(I)=XXEP1                                             PAR00220
0011                  RETURN                                                     PAR00230
0012                  END                                                        PAR00240
```

Figure 25.--Subroutine PARDIF.

```
FORTRAN IV G LEVEL  21                PERT                DATE = 80242          12/13/33

                C SUBROUTINE PERT IS CODED BY GHAEMI(1979) TO PERFORM THE PARAMETER     PER00060
                C ADJUASTMENTS REQUIRED AT VARIOUS STAGES OF BOUND CALCULATIONS.       PER00070
0001                  SUBROUTINE PERT                                            PER00080
0002                  IMPLICIT REAL*8(A-H,O-Z)                                   PER00090
0003                  COMMON/ABG/LY,LZ,PER(45)                                   PER00100
0004                  COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS                     PER00110
0005                20 LZ=LZ+1                                                   PER00120
0006                   IF(LZ .GT. NPAR) GO TO 30                                 PER00130
0007                   IF(PER(LZ) .NE. 0.) GO TO 10                              PER00140
0008                   GO TO 20                                                  PER00150
0009                10 PAR(LZ)=PAR(LZ)+PER(LZ)                                   PER00160
0010                30 RETURN                                                    PER00170
0011                   END                                                       PER00180
```

Figure 26.--Subroutine PERT.

6.19 PEVALU

Subroutine PEVALU is used to compute the value of the W function and the G function. It makes use of the values of the objective function and the constraint function, which must have been computed before PEVALU is called. The G function is interpreted as the value of the dual objective function when the W function has been minimized for a given value of r (RHØ). PEVALU is given as Figure 27.

```
FORTRAN IV G LEVEL  21                PEVALU          DATE = 80242        12/11/19

0001                SUBROUTINE PEVALU                                          PEV00040
                    C                                                          PEV00050
                    C            OCTOBER 1970                                  PEV00060
                    C                                                          PEV00070
                    C PEVALU COMPUTES THE VALUE OF THE PENALTY FUNCTION AND THE VALUE OF THEPEV00080
                    C DUAL USING PREVIOUSLY COMPUTED VALUES FOR F, AND RJ.     PEV00090
0002                IMPLICIT REAL*8(A-H,O-Z)                                   PEV00100
0003                REAL*4 RHOIN,RATIO,EPSI,THETAO                             PEV00110
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MM,NP1,NM1         PEV00120
0005                COMMON /EQAL/ H, HI, MZ                                    PEV00130
0006                COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10    PEV00140
0007                COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                      PEV00150
0008                COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,    PEV00160
                   1RSIGL,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,         PEV00170
                   2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                 PEV00180
                   3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                PEV00190
0009                H=0.0                                                      PEV00200
0010                RSIGMA=0.0                                                 PEV00210
                    C NONNEGS IF INCLUDED ARE ADDED TO  P—ARE POS. IN ALL PHASES PEV00220
0011                GO TO (10,30), NT2                                         PEV00230
0012           10   DO 20 I=1,N                                                PEV00240
0013           20   RSIGMA=RSIGMA-RHO*DLOG(X(I))                              PEV00250
0014           30   GO TO (40,50,150), NPHASE                                  PEV00260
                    C OBJECTIVE FUNCTION   - SIGMA VIOL. CONSTS.               PEV00270
0015           40   F=0.0                                                      PEV00280
0016           50   IF (M) 100,100,60                                          PEV00290
0017           60   DO 90 J=1,M                                                PEV00300
0018                IF (RJ(J)) 80,80,70                                        PEV00310
0019           70   RSIGMA=RSIGMA-RHO*DLOG(RJ(J))                             PEV00320
0020                GO TO 90                                                   PEV00330
0021           80   F=F-RJ(J)                                                  PEV00340
0022           90   CONTINUE                                                   PEV00350
                    C EQUALITIES NOT ADDED IN FEAS. PHASE                      PEV00360
0023           100  CONTINUE                                                   PEV00370
0024                IF (MZ) 140,140,110                                        PEV00380
0025           110  GO TO (140,120,150), NPHASE                                PEV00390
0026           120  DO 130 I=1,MZ                                              PEV00400
0027                K=M+I                                                      PEV00410
0028           130  H=H+RJ(K)**2                                               PEV00420
0029                H=H/RHO                                                    PEV00430
0030           140  HS=H+RSIGMA                                                PEV00440
0031                PO=F+HS                                                    PEV00450
0032                HMS=2.*H-RHO*DFLOAT(M)                                     PEV00460
0033                G=F+HMS                                                    PEV00470
0034                IF(NT2.EQ.1) G=G-RHO*DFLOAT(N)                             PEV00480
0035           150  RETURN                                                     PEV00490
0036                END                                                        PEV00500
```

Figure 27.--Subroutine PEVALU.

6.20 PRESEN

This subroutine calculates the gradient of the optimal value function, using the gradient of the Lagrangian when taken with respect to the parameters (Step 9 of Algorithm 2.1). In addition, PRESEN (when invoked) performs a preliminary screening of the problem parameters to which the optimal value function is practically insensitive. The criterion used for this purpose is that if the change in optimal value function is less than 0.1 percent of its current value as a result of the introduced increment of a given parameter, then that parameter is

eliminated from further consideration. This parameter screening is invoked when the fifth variable in the second option card takes a value of zero, as shown in Table 4. Figure 28 lists PRESEN.

6.21 PUNCH

When a call to PUNCH is made, the current value of the array x is punched, along with some of the control cards that can be used to restart SENSUMT. Subroutine PUNCH is shown in Figure 29.

6.22 REJECT

Subroutine REJECT puts values of the point at x and the associated values of the objective function, constraint functions, and W function back into their normal locations. These values were temporarily stored in other locations by subroutine STØRE. STØRE appears as Figure 30.

6.23 RHØCØM

This subroutine is used to compute the initial value of r (RHØ) for each NLP problem. As previously stated, the search for a feasible starting point can result in a sequence of NLP problems. The initial value of r is computed by the formula specified for the use of option 1 (NT1). RHØCØM is given as Figure 31.

6.24 SECØND

SECØND is used to query about the computer's clock. This is made possible by an internal clock which is called by this subroutine. The elapsed time obtained and monitored by this subroutine is used by the subroutines TIMEC, TCHECK, and SET. Figure 32 is subroutine SECØND.

```
FORTRAN IV G LEVEL 21          PRESEN          DATE = 80242          12/13/50

0001            SUBROUTINE PRESEN(DU,KTEST)                                    PRE00060
                C                                                              PRE00070
                C                    1 MARCH 1976                              PRE00080
                C                                                              PRE00090
                C  SUBROUTINE PRESEN IS USED TO CALCULATE THE GRADIENT OF THE OPTIMAL   PRE00100
                C  VALUE FUNCTION USING THE GRADIENT OF THE LAGRANGIAN TAKEN WITH       PRE00110
                C  RESPECT TO THE PARAMETERS.  PRESEN IS CALLED BY SENS WHEN VARIABLE   PRE00120
                C  NEXOPS =0 OR =1.  ADDITIONALLY, WHEN NEXOPS = 0, THE PARAMETERS WHICHPRE00130
                C  AFFECT THE OPTIMAL VALUE FUNCTION BY LESS THAN 0.001 TIMES ITS      PRE00140
                C  CURRENT VALUE ARE ELIMINATED FROM FURTHER CONSIDERATION.  THIS      PRE00150
                C  SUBROUTINE WAS CODED BY R. L. ARMACOST.                            PRE00160
0002                IMPLICIT REAL*8(A-H,O-Z)                                   PRE00170
0003                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1         PRE00180
0004                COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                      PRE00190
0005                COMMON/EQAL/ M, M1,MZ                                      PRE00200
0006                COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS                     PRE00210
0007                COMMON/ABG/LY,LZ,PER(45)                                   PRE00220
0008                COMMON/ABG1/FE1,FE2                                        PRE00230
0009                COMMON/ABG2/DF1(45),DF2(45)                                PRE00240
0010                DIMENSION GX(90),DU(90),KTEST(45),KLIST(45)                PRE00250
0011                MPMZ = M + MZ                                              PRE00260
0012                FTEST = 0.001 * DABS(F)                                    PRE00270
0013                DO 100 J=1,NPAR                                            PRE00280
0014                KTEST(J) = 0                                               PRE00290
0015                PAR(J) = PAR(J) + DPAR(J)                                  PRE00300
0016                CALL RESTNT(0,DF)                                          PRE00310
0017                IF(MPMZ.EQ.0) GO TO 20                                     PRE00320
0018                DO 10 I=1,MPMZ                                             PRE00330
0019                CALL RESTNT(I,DU(I))                                       PRE00340
0020           10   CONTINUE                                                   PRE00350
0021           20   DEM = 2. * DPAR(J)                                         PRE00360
0022                PAR(J) = PAR(J) - DEM                                      PRE00370
0023                CALL RESTNT(0,XF)                                          PRE00380
0024                IF(MPMZ.EQ.0) GO TO 40                                     PRE00390
0025                DO 30 I=1,MPMZ                                             PRE00400
0026                CALL RESTNT(I,GX(I))                                       PRE00410
0027           30   CONTINUE                                                   PRE00420
0028           40   DFEPS = (DF - XF)/DEM                                      PRE00430
0029                IF(MPMZ.EQ.0) GO TO 60                                     PRE00440
0030                DO 50 I=1,MPMZ                                             PRE00450
0031           50   DU(I) = (DU(I) - GX(I))/DEM                                PRE00460
0032           60   SUM = DFEPS                                                PRE00470
0033                IF(M.EQ.0) GO TO 80                                        PRE00480
0034                DO 70 I=1,M                                                PRE00490
0035           70   SUM = SUM - RHO/RJ(I)*DU(I)                                PRE00500
0036           80   IF(MZ.EQ.0) GO TO 95                                       PRE00510
0037                TSUM = 0.                                                  PRE00520
0038                DO 90 I=1,MZ                                               PRE00530
0039                IM = I+M                                                   PRE00540
0040           90   TSUM = TSUM + RJ(IM)*DU(IM)                                PRE00550
0041                SUM = SUM + TSUM * 2./RHO                                  PRE00560
0042           95   DEL(J) = SUM                                               PRE00570
0043                PAR(J) = PAR(J) + DPAR(J)                                  PRE00580
0044                DTEST = DABS(DEL(J))                                       PRE00590
0045                IF(DTEST.GE.FTEST) KTEST(J) = 1                            PRE00600
0046           100  CONTINUE                                                   PRE00610
0047                WRITE(6,600)                                               PRE00620
0048           600  FORMAT(22X,34HOPTIMAL VALUE FUNCTION SENSITIVITY     //)  PRE00630
0049                DO 200 I=1,NPAR,5                                          PRE00640
0050                II=MINO(I+4,NPAR)                                          PRE00650
0051                WRITE(6,601) ((JJ,DEL(JJ)), JJ=I,II)                       PRE00660
0052           601  FORMAT( 5(7H DF/DA(,I2,2H)=,G14.7))                        PRE00670
0053           200  CONTINUE                                                   PRE00680
0054                IF(LY .GT. 0.) GO TO 500                                   PRE00690
0055                DO 400 J=1,NPAR                                            PRE00700
0056                DF1(J)=DEL(J)                                              PRE00710
0057           400  CONTINUE                                                   PRE00720
0058                GO TO 700                                                  PRE00730
0059           500  DO 610 J=1,NPAR                                            PRE00740
0060                DF2(J)=DEL(J)                                              PRE00750
0061           610  CONTINUE                                                   PRE00760
0062                IF(LY .EQ. 0.) GO TO 700                                   PRE00770
0063                PAR(LZ)=PAR(LZ)-PER(LZ)                                    PRE00780
0064                RETURN                                                     PRE00790
0065           700  JJ = 0                                                     PRE00800
0066                DO 250 J=1,NPAR                                            PRE00810
0067                IF(KTEST(J).EQ.0) GO TO 250                                PRE00820
0068                JJ = JJ + 1                                                PRE00830
0069                KLIST(JJ) = J                                              PRE00840
0070           250  CONTINUE                                                   PRE00850
0071                IF(JJ.EQ.0) GO TO 300                                      PRE00860
0072                WRITE(6,602)                                               PRE00870
0073           602  FORMAT( /51H DETAILED SENSITIVITY RESULTS FOLLOW FOR PARAMETERS )PRE00880
0074                WRITE(6,603) (KLIST(I), I=1,JJ)                            PRE00890
0075           603  FORMAT(1H 40(I2,2H ,))                                     PRE00900
0076                WRITE(6,604)                                               PRE00910
0077           604  FORMAT(/)                                                  PRE00920
0078                RETURN                                                     PRE00930
0079           300  WRITE(6,605)                                               PRE00940
0080           605  FORMAT( 42H THERE ARE NO DETAILED SENSITIVITY RESULTS  //) PRE00950
0081                RETURN                                                     PRE00960
0082                END                                                        PRE00970
```

Figure 28.--Subroutine PRESEN.

```
FORTRAN IV G LEVEL  21                PUNCH           DATE = 80242        12/12/21

0001                SUBROUTINE PUNCH                                                PUN00040
                C                                                                   PUN00050
                C              OCTOBER 1970                                         PUN00060
                C                                                                   PUN00070
                C THIS SUBROUTINE PUNCHES THE STOPPING POINTS AND ASSOCIATED PARAMETERS PUN00080
                C SO THAT ANOTHER RUN MAY BE MADE STARTING WHERE THE CURRENT ONE    PUN00090
                C STOPPED                                                           PUN00100
                C  THIS ROUTINE IS CALLED IF NT6=2.                                 PUN00110
0002                IMPLICIT REAL*8(A-H,O-Z)                                        PUN00120
0003                REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2,T                      PUN00130
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              PUN00140
0005                COMMON /EQAL/ H, H1, MZ                                         PUN00150
0006                COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10         PUN00160
0007                COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                           PUN00170
0008                COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         PUN00180
                   1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,              PUN00190
                   2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      PUN00200
                   3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                     PUN00210
0009                COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2      PUN00220
0010                T=60.0                                                          PUN00230
0011                WRITE (7,10) EPSI,RHO,THETAO,RATIO,T,M,N,MZ                     PUN00240
                C  TMMAX.IS SET TO 60. SECONDS                                      PUN00250
0012                WRITE (7,20) (X(I),I=1,N)                                       PUN00260
0013                NT1=3                                                           PUN00270
                C SET  RHO OPTION SO THIS VALUE OF RHO WILL BE USE FOR THE RESTART. PUN00280
0014                WRITE (7,30) NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10           PUN00290
0015                WRITE (7,20) XEP1, XEP2                                         PUN00300
0016                WRITE (7,30) NEXOP1,NEXOP2,NEXOP3                               PUN00310
0017                RETURN                                                          PUN00320
                C                                                                   PUN00330
0018            10  FORMAT (5E12.5,3I4)                                            PUN00340
0019            20  FORMAT (6E12.5)                                                PUN00350
0020            30  FORMAT (10I7)                                                  PUN00360
0021                END                                                             PUN00370
```

Figure 29.--Subroutine PUNCH.

```
FORTRAN IV G LEVEL  21                REJECT          DATE = 80242        12/13/03

0001                SUBROUTINE REJECT                                               REJ00040
                C                                                                   REJ00050
                C              OCTOBER 1970                                         REJ00060
                C                                                                   REJ00070
                C REJECT RETURNS THE STORED VALUES OF THE OBJECTIVE FUNCTION, THE   REJ00080
                C CONSTRAINT FUNCTIONS AND THE PENALTY FUNCTION TO THEIR NORMAL LOCATIONREJ00090
0002                IMPLICIT REAL*8(A-H,O-Z)                                        REJ00100
0003                REAL*4 RHOIN,RATIO,EPSI,THETAO                                  REJ00110
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              REJ00120
0005                COMMON /EQAL/ H, H1, MZ                                         REJ00130
0006                COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                           REJ00140
0007                COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         REJ00150
                   1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,              REJ00160
                   2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      REJ00170
                   3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                     REJ00180
0008                DO 10 I=1,N                                                     REJ00190
0009            10  X(I)=X1(I)                                                     REJ00200
0010                MMZ=M+MZ                                                        REJ00210
0011                DO 20 J=1,MMZ                                                   REJ00220
0012            20  RJ(J)=RJ1(J)                                                   REJ00230
0013                PO=P1                                                           REJ00240
0014                RSIGMA=RSIG1                                                    REJ00250
0015                G=G1                                                            REJ00260
0016                F=F1                                                            REJ00270
0017                H=H1                                                            REJ00280
0018                RETURN                                                          REJ00290
0019                END                                                             REJ00300
```

Figure 30.--Subroutine REJECT.

```
FURTRAN IV G LEVEL  21                  RHOCOM              DATE = 80242          12/39/21

0001                  SUBROUTINE RHOCOM                                                RHO00040
                C                                                                      RHO00050
                C                 OCTOBER 1970                                         RHO00060
                C                                                                      RHO00070
                C     SUBROUTINE  TO COMPUTE INITIAL RHO VALUE                         RHO00080
                C     CONTROLLED  BY  COL. 7  ON OPTION CARD                           RHO00090
0002                  IMPLICIT REAL*8(A-H,O-Z)                                         RHO00100
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                   RHO00110
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1               RHO00120
0005                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10          RHO00130
0006                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                            RHO00140
0007                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,          RHO00150
                     1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,              RHO00160
                     2PR2,P1,F1,RJ1(90),DOTF,PGRAD(45),DIAG(45),                      RHO00170
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                     RHO00180
0008                  GO TO (110,50,10,190), NT1                                       RHO00190
0009            10    RHO=RHOIN                                                        RHO00200
0010            20    IF (RHO) 30,30,40                                                RHO00210
0011            30    RHO=1.                                                           RHO00220
0012            40    RETURN                                                           RHO00230
0013            50    NPAR1=1                                                          RHO00240
0014            60    RHO=1.                                                           RHO00250
                C     2 MEANS  RHO WHICH  MINIMIZES GRADIENT  MAG.                     RHO00260
0015                  CALL GRAD (2)                                                    RHO00270
0016                  DO 70 I=1,N                                                      RHO00280
0017            70    PGRAD(I)=DELXO(I)                                                RHO00290
0018                  RHO=2.                                                           RHO00300
0019                  CALL GRAD (2)                                                    RHO00310
0020                  DO 80 I=1,N                                                      RHO00320
0021                  DELXO(I)=DELXO(I)-PGRAD(I)                                       RHO00330
0022            80    PGRAD(I)=PGRAD(I)-DELXO(I)                                       RHO00340
0023                  GO TO (90,130), NPAR1                                            RHO00350
0024            90    DOT1=0.                                                          RHO00360
0025                  DOT2=0.                                                          RHO00370
0026                  DO 100 I=1,N                                                     RHO00380
0027                  DOT1=DOT1+DELXO(I)*PGRAD(I)                                      RHO00390
0028            100   DOT2=DOT2+DELXO(I)**2                                            RHO00400
0029                  RHO=DABS(DOT1/DOT2)                                              RHO00410
0030                  GO TO 20                                                         RHO00420
                C  3 MEANS COMPUTE RHO SO AS TO MINIMIZE DEL P1/DDP/1.)DEL P           RHO00430
0031            110   NPAR2=1                                                          RHO00440
0032            120   NPAR1=2                                                          RHO00450
                C     USE  DF  AND DR  SUBROUTINE                                      RHO00460
0033                  GO TO 60                                                         RHO00470
0034            130   RHO=1.                                                           RHO00480
                C     ASSUME   SIGMA TERM IS CONSID. GRTER THAN  F TERM                RHO00490
0035                  CALL SECORD (2)                                                  RHO00500
0036                  DO 140 I=1,N                                                     RHO00510
0037            140   DELX(I)=PGRAD(I)                                                 RHO00520
0038                  CALL INVERS (1)                                                  RHO00530
0039                  DO 150 I=1,N                                                     RHO00540
0040                  X1(I)=DELX(I)                                                    RHO00550
0041            150   DELX(I)=DELXO(I)                                                 RHO00560
0042                  CALL SECORD (2)                                                  RHO00570
0043                  CALL INVERS (1)                                                  RHO00580
0044                  DO 160 I=1,N                                                     RHO00590
0045            160   XR2(I)=DELX(I)                                                   RHO00600
0047            170   DOT1=0.                                                          RHO00620
0048                  DOT2=0.                                                          RHO00630
0049                  DO 180 I=1,N                                                     RHO00640
0050                  DOT1=DOT1+PGRAD(I)*X1(I)                                         RHO00650
0051            180   DOT2=DOT2+DELXO(I)*XR2(I)                                        RHO00660
0052                  RHO=DSQRT(DABS(DOT1/DOT2))                                       RHO00670
0053                  GO TO 20                                                         RHO00680
0054            190   NPAR2=2                                                          RHO00690
                C     RHO MINIMIZES  2ND ORDER MOVE                                    RHO00700
0055                  GO TO 120                                                        RHO00710
                C     USES  INTERNAL  SUB. TO COM  /DDP/-1 DF AND  /DDP/- DR           RHO00720
0056            200   DOT1=0.0                                                         RHO00730
0057                  DOT2=0.0                                                         RHO00740
0058                  DO 210 I=1,N                                                     RHO00750
0059                  DOT1=X1(I)**2+DOT1                                               RHO00760
0060            210   DOT2=X1(I)*XR2(I)+DOT2                                           RHO00770
0061                  RHO=DABS(DOT1/DOT2)                                              RHO00780
0062                  GO TO 20                                                         RHO00790
0063                  END                                                              RHO00800
```

Figure 31.--Subroutine RHØCØM.

```
FORTRAN IV G LEVEL 21          SECUND          DATE = 80242          12/14/02

0001          SUBROUTINE SECOND(SECS)                                SEC00050
0002          CALL CLOCK(I)                                          SEC00060
0003          SECS = I                                               SEC00070
0004          SECS = SECS / 100.0                                    SEC00080
0005          RETURN                                                 SEC00090
0006          END                                                    SEC00100
```

Figure 32.--Subroutine SECØND.

6.25 SECØRD

Subroutine SECØRD evaluates the matrix of second partials of the W function. It calls on the user-supplied subroutine MATRIX to evaluate the upper triangle of the matrix of the second partials of the objective and the constraint functions. Subroutine SECØRD is listed in Figure 33.

6.26 SENS

This subroutine calculates the solution point and optimal value function sensitivities for each subproblem (if required), as well as the final problem, by implementing Steps 1-3 and 7-8 of Algorithm 2.1 of Section 2. The various choices for sensitivity calculations are given in options 3, 4, and 5 of the second option card (see Table 4). SENS is called by the program MAIN and the subroutine BØDY. It calls subroutine LMULT to calculate the Lagrange multiplier sensitivities. SENS is shown as Figure 34.

6.27 SET

Subroutine SET is called once by MAIN at the start of the attempt to solve a problem. It obtains and stores the value of the computer's clock. It obtains the value of the clock by calling a system library or user coded subroutine that queries the computer's clock and returns its value as a floating point number in seconds. Figure 35 lists subroutine SET.

```
FORTRAN IV G LEVEL 21                SECORD              DATE = 80242          12/46/12

0001                  SUBROUTINE SECORD (IS)                                          SEC00040
                C                                                                     SEC00050
                C              OCTOBER 1970                                           SEC00060
                C                                                                     SEC00070
                C SECORD EVALUATES THE MATRIX OF SECOND PARTIALS OF THE PENALTY        SEC00080
                C FUNCTION.                                                           SEC00090
                C                                                                     SEC00100
                C- (1) MEANS DONT   COMPUTE GRAD. OUTER PRODUCT(IN SECORD)            SEC00110
0002                  IMPLICIT REAL*8(A-H,O-Z)                                        SEC00120
0003                  REAL*4 RHOIN,RATIO,EPSI,THETAO                                  SEC00130
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              SEC00140
0005                  COMMON /EQAL/ H, H1, MZ                                         SEC00150
0006                  COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10         SEC00160
0007                  COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                           SEC00170
0008                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         SEC00180
                     1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,              SEC00190
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      SEC00200
                     3 PREV3,ADELX, NTCTR, NUMINI  NPHASE, NSATIS                     SEC00210
0009                  DO 10 I=1,N                                                     SEC00220
0010                  DO 10 J=1,N                                                     SEC00230
0011            10    A(I,J)=0.                                                       SEC00240
0012                  GO TO (230,20), IS                                              SEC00250
                C GRAD. TERM NOT PREV. COMPUTED                                       SEC00260
0013            20    DO 30 I=1,N                                                     SEC00270
0014                  DO 30 J=1,I                                                     SEC00280
0015                  A(I,J)=0.0                                                      SEC00290
0016            30    CONTINUE                                                        SEC00300
0017                  GO TO (40,60), NT2                                              SEC00310
0018            40    DO 50 I=1,N                                                     SEC00320
0019            50    A(I,I)=RHO/X(I)**2                                              SEC00330
0020            60    CONTINUE                                                        SEC00340
0021                  IF (M.LE.0) GO TO 130                                           SEC00350
0022                  DO 120 IN=1,M                                                   SEC00360
0023                  IF (RJ(IN)) 120,120,70                                          SEC00370
0024            70    CALL GRAD1 (IN)                                                 SEC00380
0025                  TT = RHO/RJ(IN)**2                                              SEC00390
0026                  DO 110 I=1,N                                                    SEC00400
0027                  IF (DEL(I)) 80,110,80                                           SEC00410
0028            80    T=TT*DEL(I)                                                     SEC00420
0029                  DO 100 J=1,I                                                    SEC00430
0030                  IF (DEL(J)) 90,100,90                                           SEC00440
0031            90    A(I,J)=A(I,J)+T*DEL(J)                                          SEC00450
0032           100    CONTINUE                                                        SEC00460
0033           110    CONTINUE                                                        SEC00470
0034           120    CONTINUE                                                        SEC00480
                C EQUALITY  CONSTRAINTS                                               SEC00490
0035           130    IF (MZ) 210,210,140                                             SEC00500
0036           140    GO TO (210,150,230), NPHASE                                     SEC00510
0037           150    RQ=2./RHO                                                       SEC00520
0038                  DO 200 JJ=1,MZ                                                  SEC00530
0039                  IN=M+JJ                                                         SEC00540
0040                  CALL GRAD1 (IN)                                                 SEC00550
0041                  DO 190 I=1,N                                                    SEC00560
0042                  IF (DEL(I)) 160,190,160                                         SEC00570
0043           160    T=RQ*DEL(I)                                                     SEC00580
0044                  DO 180 J=1,I                                                    SEC00590
0045                  IF (DEL(J)) 170,180,170                                         SEC00600
0046           170    A(I,J)=A(I,J)+T*DEL(J)                                          SEC00610
0047           180    CONTINUE                                                        SEC00620
0048           190    CONTINUE                                                        SEC00630
0049           200    CONTINUE                                                        SEC00640
0050           210    DO 220 I=1,N                                                    SEC00650
0051                  DIAG(I)=A(I,I)                                                  SEC00660
0052           220    A(I,I)=0.                                                       SEC00670
                C  READY NOW FOR MATRIX OF 2ND PARTIALS OF RESTRAINTS                 SEC00680
0053           230    GO TO (240,510,520), NT10                                       SEC00690
0054           240    IF (M.LE.0) GO TO 340                                           SEC00700
0055                  DO 330 IN=1,M                                                   SEC00710
0056                  LORN=2                                                          SEC00720
                C CONSTRAINT ASSUMED NONLINEAR                                        SEC00730
0057                  CALL MATRIX (IN,LORN)                                           SEC00740
0058                  IF (LORN.LT.2) GO TO 330                                        SEC00750
0059                  IF (RJ(IN) .GT. 0.0) GO TO 280                                  SEC00760
0060                  DO 260 I=2,N                                                    SEC00770
0061                  IM1=I-1                                                         SEC00780
0062                  DO 260 J=1,IM1                                                  SEC00790
0063                  IF (A(J,I)) 250,260,250                                         SEC00800
0064           250    A(I,J)=A(I,J)-A(J,I)                                            SEC00810
0065                  A(J,I)=0.                                                       SEC00820
0066           260    CONTINUE                                                        SEC00830
0067                  DO 270 I=1,N                                                    SEC00840
0068                  DIAG(I)=DIAG(I)-A(I,I)                                          SEC00850
0069           270    A(I,I)=0.0                                                      SEC00860
0070                  GO TO 330                                                       SEC00870
0071           280    T=-RHO/RJ(IN)                                                   SEC00880
0072                  DO 300 I=2,N                                                    SEC00890
0073                  IM1=I-1                                                         SEC00900
```

Figure 33.--Subroutine SECØRD.

```
0074              DO 300 J=1,IM1                                          SEC00910
0075              IF (A(J,I)) 290,300,290                                 SEC00920
0076        290   A(I,J)=A(I,J)+T*A(J,I)                                  SEC00930
0077              A(J,I)=0.                                               SEC00940
0078        300   CONTINUE                                                SEC00950
0079              DO 320 I=1,N                                            SEC00960
0080              IF (A(I,I)) 310,320,310                                 SEC00970
0081        310   DIAG(I)=DIAG(I)+T*A(I,I)                                SEC00980
0082              A(I,I)=0.                                               SEC00990
0083        320   CONTINUE                                                SEC01000
0084        330   CONTINUE                                                SEC01010
0085        340   CONTINUE                                                SEC01020
0086              GO TO (520,350,520), NPHASE                             SEC01030
0087        350   IF (MZ.EQ.0) GO TO 420                                  SEC01040
                  C--    EQUALITY  SECOND PARTIALS HERE                   SEC01050
0088              IF (NT10.GE.2) GO TO 420                                SEC01060
0089              DO 410 II=1,MZ                                          SEC01070
0090              IN=M+II                                                 SEC01080
0091              LORN=2                                                  SEC01090
0092              CALL MATRIX (IN,LORN)                                   SEC01100
0093              IF (LORN.LT.2) GO TO 410                                SEC01110
0094              T=2.*RJ(IN)/RHO                                         SEC01120
0095              DO 380 I=2,N                                            SEC01130
0096              IM1=I-1                                                 SEC01140
0097              DO 370 J=1,IM1                                          SEC01150
0098              IF (A(J,I)) 360,370,360                                 SEC01160
0099        360   A(I,J)=A(I,J)+T*A(J,I)                                  SEC01170
0100              A(J,I)=0.0                                              SEC01180
0101        370   CONTINUE                                                SEC01190
0102        380   CONTINUE                                                SEC01200
0103              DO 400 I=1,N                                            SEC01210
0104              IF (A(I,I)) 390,400,390                                 SEC01220
0105        390   DIAG(I)=DIAG(I)+T*A(I,I)                                SEC01230
0106              A(I,I)=0.0                                              SEC01240
0107        400   CONTINUE                                                SEC01250
0108        410   CONTINUE                                                SEC01260
                  C  GET MATRIX OF 2ND PARTIALS OF OBJECTIVE FUNCTION     SEC01270
0109        420   LLL=2                                                   SEC01280
0110              CALL MATRIX (0,LLL)                                     SEC01290
0111              IF (LLL.LT.2) GO TO 490                                 SEC01300
0112              DO 440 I=2,N                                            SEC01310
0113              IM1=I-1                                                 SEC01320
0114              DO 440 J=1,IM1                                          SEC01330
0115              IF (A(J,I)) 430,440,430                                 SEC01340
0116        430   A(I,J)=A(I,J)+A(J,I)                                    SEC01350
0117        440   A(J,I)=A(I,J)                                           SEC01360
0118              DO 470 I=1,N                                            SEC01370
0119              IF (A(I,I)) 450,460,450                                 SEC01380
0120        450   A(I,I)=DIAG(I)+A(I,I)                                   SEC01390
0121              GO TO 470                                               SEC01400
0122        460   A(I,I)=DIAG(I)                                          SEC01410
0123        470   CONTINUE                                                SEC01420
0124        480   RETURN                                                  SEC01430
0125        490   DO 500 I=1,N                                            SEC01440
0126              A(I,I)=DIAG(I)                                          SEC01450
0127              DO 500 J=1,N                                            SEC01460
0128        500   A(I,J)=A(J,I)                                           SEC01470
0129              GO TO 480                                               SEC01480
0130        510   GO TO (520,350,350), NPHASE                             SEC01490
0131        520   DO 530 I=2,N                                            SEC01500
0132              IM1=I-1                                                 SEC01510
0133              DO 530 J=1,IM1                                          SEC01520
0134        530   A(J,I)=A(I,J)                                           SEC01530
0135              DO 540 I=1,N                                            SEC01540
0136        540   A(I,I)=DIAG(I)                                          SEC01550
0137              GO TO 480                                               SEC01560
0138              END                                                     SEC01570
```

Figure 33.--*continued*

```
FORTRAN IV G LEVEL 21          SENS          DATE = 80242          12/27/55

0001            SUBROUTINE SENS                                                 SEN00060
                C                                                               SEN00070
                C                    1 MARCH 1976                               SEN00080
                C                                                               SEN00090
                C  THIS VERSION OF THE SENSITIVITY ANALYSIS SUBROUTINE IS USED TO       SEN00100
                C  COMPUTE THE DIRECTIONAL DERIVATIVES OF X AND F WITH RESPECT TO       SEN00110
                C  CERTAIN PARAMETERS CODED IN THE ARRAY PAR(20).  THE DIRECTIONAL      SEN00120
                C  DERIVATIVES ARE ESTIMATED FOR ONE PARAMETER AT A TIME WITH NPAR BEINGSEN00130
                C  THE NUMBER OF PARAMETERS INVOLVED IN THE SENSITIVITY ANALYSIS.  THE  SEN00140
                C  USE OF THE PARAMETERS PAR(20) MUST BE CONSISTENT THROUGHOUT THE      SEN00150
                C  USER'S SUBROUTINES.                                                 SEN00160
                C  THE SUBROUTINE IS USED FOR A SENSITIVITY ANALYSIS AT THE FINAL SUB-  SEN00170
                C  PROBLEM OR FOR A SENSITIVITY ANALYSIS AT EACH SUBPROBLEM ALONG THE   SEN00180
                C  MINIMIZING TRAJECTORY.  DPAR(20) IS THE ARRAY OF DIFFERENCING       SEN00190
                C  INTERVALS CORRESPONDING TO THE PARAMETERS PAR(20).  DPAR(20) IS     SEN00200
                C  ASSIGNED VALUES IN SUBROUTINE PARDIF.                               SEN00210
                C  THIS APPROACH TO SENSITIVITY ANALYSIS IS DUE TO A. V. FIACCO. THE   SEN00220
                C  FIRST VERSION WAS CODED BY B. CAUSEY.  THE SECOND VERSION WAS CODED SEN00230
                C  BY W. C. MYLANDER.  THE THIRD VERSION WAS AN EXTENSION OF THE       SEN00240
                C  SECOND VERSION TO PERMIT SENSITIVITY ANALYSIS ALONG THE MINIMIZING  SEN00250
                C  TRAJECTORY, AND WAS CODED BY R. L. ARMACOST.  THIS IS THE FOURTH    SEN00260
                C  VERSION WHICH INCORPORATES THE OPTION TO CALCULATE THE PARTIAL       SEN00270
                C  DERIVATIVES OF THE LAGRANGE MULTIPLIERS (SUBROUTINE LMULT) AND TO   SEN00280
                C  CONDUCT A PRELIMINARY SCREENING OF THE PARAMETERS BY ESTIMATING THE SEN00290
                C  FIRST ORDER SENSITIVITY OF THE OPTIMAL VALUE FUNCTION (SUBROUTINE    SEN00300
                C  PRESEN).  THIS VERSION WAS CODED BY R. L. ARMACOST.                 SEN00310
                C THIS ROUTINE TOGETHER WITH OTHER SENSITIVITY ROUTINE WERE             SEN00320
                C REDIMENSIONED TO 45 I.E. X(45) AND PAR(45). FURTHERMORE              SEN00330
                C IT WAS SLIGHTLY MODIFIED TO INTERFACE BOUND CALCULATION              SEN00340
                C ROUTINES. GHAEMI(1979).                                              SEN00350
0002                IMPLICIT REAL*8(A-H,O-Z)                                        SEN00360
0003                REAL*4 RHOIN,RATIO,EPSI,THETAO,XEP1,XEP2                        SEN00370
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              SEN00380
0005                COMMON/EQAL/H,H1,MZ                                             SEN00390
0006                COMMON/OPTNS/NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10           SEN00400
0007                COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                           SEN00410
0008                COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         SEN00420
                   1NTCTR,NUMINI,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,          SEN00430
                   2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      SEN00440
                   3PREV3,ADELX,RSIG1,G1,NPHASE,NSATIS                              SEN00450
0009                COMMON/SEN/PAR(45),DPAR(45),NPAR,ISENS                          SEN00460
0010                COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XEP1,XEP2      SEN00470
0011                COMMON/ABG/LY,LZ,PER(45)                                        SEN00480
0012                COMMON/ABG1/FE1,FE2                                             SEN00490
0013                COMMON/ABG2/DF1(45),DF2(45)                                     SEN00500
0014                COMMON/ABG5/TX(45),TDELX(45)                                    SEN00510
0015                COMMON/OP6/NEXOP6                                               SEN00520
0016                DIMENSION DELT(45),DELTX(45),X2H(45),X3H(45)                    SEN00530
0017                DIMENSION DELMU(90),DU(90),KTEST(45)                            SEN00540
0018                DO 5 I=1,N                                                      SEN00550
0019                DELTX(I) = DELX(I)                                              SEN00560
0020              5 DELT(I) = DELXO(I)                                              SEN00570
0021                CALL STORE                                                      SEN00580
0022                AVAL = 1.0E-10                                                  SEN00590
0023                MPMZ = M + MZ                                                   SEN00600
0024                ISENS = ISENS + 1                                               SEN00610
0025                CALL PARDIF                                                     SEN00620
                C  WRITE OUT X, RHO AND PAR.                                       SEN00630
0026                WRITE(6,10) RHO                                                 SEN00640
0027               10 FORMAT(1H //30X,20HSENSITIVITY ANALYSIS    //                 SEN00650
                     15X,22HTHE VALUE OF R(RHO) IS ,E12.5)                         SEN00660
0028                WRITE(6,20)                                                     SEN00670
0029               20 FORMAT(/6X, 64HTHE POINT AT WHICH THE ESTIMATE OF SENSITIVITY WILLSEN00680
                     1BE MADE IS     )                                             SEN00690
0030                DO 40 I=1,N,6                                                   SEN00700
0031                II=MIN0(I+5,N)                                                  SEN00710
0032                WRITE(6,30) ((J,X(J)), J=I,II)                                  SEN00720
0033               30 FORMAT( 6(2X,2HX(,I2,2H)=,G14.7) )                            SEN00730
0034               40 CONTINUE                                                      SEN00740
                C CALCULATION OF GP VARIABLE VALUE                                 SEN00750
0035                IF(NEXOP6.NE.1) GO TO 660                                       SEN00760
0036                INDEX=0.                                                        SEN00770
0037                CALL TRANS(INDEX)                                               SEN00780
0038                WRITE(6,600)                                                    SEN00790
0039              600 FORMAT(16X,'CORRESPONDING POINT FOR GP PROBLEM IS')           SEN00800
0040                DO 640 I=1,N,6                                                  SEN00810
0041                II=MIN0(I+5,N)                                                  SEN00820
0042              640 WRITE(6,30)((J,X(J)),J=I,II)                                  SEN00830
0043                DO 650 I=1,N                                                    SEN00840
0044              650 X(I)=TX(I)                                                    SEN00850
0045              660 CONTINUE                                                     SEN00860
0046                WRITE(6,50) ((I,PAR(I),DPAR(I)), I=1,NPAR)                      SEN00870
0047               50 FORMAT(45HO     PARAMETER VALUE   DIFFERENCING INTERVAL /    SEN00880
                     1 (I4,2X,G14.6,6X,G14.5) )                                    SEN00890
0048                WRITE(6,55)                                                     SEN00900
0049               55 FORMAT(//)                                                   SEN00910
                C  EVALUATE F, G AND H.                                            SEN00920
```

Figure 34.--Subroutine SENS.

```
0050            CALL RESTNT(0,F)                                          SEN00930
0051            IF(MPMZ.EQ.0) GO TO 85                                    SEN00940
0052            DO 80 J=1,MPMZ                                            SEN00950
0053            CALL RESTNT(J,RJ(J))                                      SEN00960
0054         80 CONTINUE                                                  SEN00970
0055            IF(NEXOP5.EQ.2) GO TO 85                                  SEN00980
0056            CALL PRESEN(DU,KTEST)                                     SEN00990
0057            IF(LY.GT.0) GO TO 400                                     SEN01000
                C  COMPUTE DEL F.                                         SEN01010
0058         85 CALL GRAD1(0)                                             SEN01020
0059            DO 90 I=1,N                                               SEN01030
0060            X3H(I) = DEL(I)                                           SEN01040
0061         90 CONTINUE                                                  SEN01050
                C  COMPUTE (DEL)**2 P - STORED IN A.                      SEN01060
0062            CALL SECOND(2)                                            SEN01070
                C  PERFORM THE L-U DECOMPOSITION OF A.                    SEN01080
0063            DO 100 I=1,N                                              SEN01090
0064            DELX(I)=0.0                                               SEN01100
0065         100 CONTINUE                                                 SEN01110
0066            NP=NT3                                                    SEN01120
0067            NT3=1                                                     SEN01130
0068            CALL INVERS(1)                                            SEN01140
                C  CHECK TO MAKE SURE AN ORTHOGONAL MOVE IS NOT ATTEMPTED. SEN01150
0069            DO 110 I=1,N                                              SEN01160
0070            IF(DELX(I).EQ.0.0) GO TO 110                              SEN01170
0071            WRITE(6,105)                                              SEN01180
0072         105 FORMAT(94HO THE MATRIX OF SECOND PARTIALS IS NOT POSITIVE DEFINITESEN01190
                   1, SENSITIVITY ANALYSIS IS TERMINATED    )             SEN01200
0073            GO TO 202                                                 SEN01210
0074         110 CONTINUE                                                 SEN01220
0075            DO 120 I=1,NPAR                                           SEN01230
0076            X2H(I) = PAR(I)                                           SEN01240
0077         120 CONTINUE                                                 SEN01250
0078            DO 200 J=1,NPAR                                           SEN01260
0079            IF(NEXOP5.NE.0) GO TO 115                                 SEN01270
0080            IF(KTEST(J).EQ.0) GO TO 200                               SEN01280
0081         115   IF(NEXOP4.EQ.0) GO TO 121                            SEN01290
0082            CALL LMULT(0,DELMU,DEM,DU)                                SEN01300
                C  COMPUTE D(DEL P)/DA(J) AND D(F)/DA(J) USING CENTRAL DIFFERENCING. SEN01310
0083         121 PAR(J) =PAR(J) + DPAR(J)                                 SEN01320
0084            CALL RESTNT(0,DF)                                         SEN01330
0085            IF(M.EQ.0) GO TO 126                                      SEN01340
0086            DO 125 I=1,M                                              SEN01350
0087            CALL RESTNT(I,RJ(I))                                      SEN01360
0088            IF(RJ(I).GT.AVAL) GO TO 125                               SEN01370
0089         123 DPAR(J)=0.1*DPAR(J)                                      SEN01380
0090            PAR(J) = X2H(J)                                           SEN01390
0091            WRITE(6,124) J,DPAR(J)                                    SEN01400
0092         124 FORMAT(16H RESETTING DPAR(,I2,3H)= ,G14.5)               SEN01410
0093            IF(DPAR(J).EQ.0.) GO TO 201                               SEN01420
0094            GO TO 121                                                 SEN01430
0095         125 CONTINUE                                                 SEN01440
0096         126 IF(MZ.EQ.0) GO TO 128                                    SEN01450
0097            DO 127 I=1,MZ                                             SEN01460
0098            MZPI=M+I                                                  SEN01470
0099            CALL RESTNT(MZPI,RJ(MZPI))                                SEN01480
0100         127 CONTINUE                                                 SEN01490
0101         128   IF(NEXOP4.EQ.0) GO TO 129                            SEN01500
0102            CALL LMULT(1,DELMU,DEM,DU)                                SEN01510
0103         129   CALL GRAD(2)                                          SEN01520
0104            DO 130 I=1,N                                              SEN01530
0105            DELX(I)=DELX0(I)                                          SEN01540
0106         130 CONTINUE                                                 SEN01550
0107            DEM=2.0*DPAR(J)                                           SEN01560
0108            PAR(J)=PAR(J) - DEM                                       SEN01570
0109            CALL RESTNT(0,VAL)                                        SEN01580
0110            IF(M.EQ.0) GO TO 136                                      SEN01590
0111            DO 135 I=1,M                                              SEN01600
0112            CALL RESTNT(I,RJ(I))                                      SEN01610
0113            IF(RJ(I).GT.AVAL) GO TO 135                               SEN01620
0114            GO TO 123                                                 SEN01630
0115         135 CONTINUE                                                 SEN01640
0116         136 IF(MZ.EQ.0) GO TO 138                                    SEN01650
0117            DO 137 I=1,MZ                                             SEN01660
0118            MZPI=M+I                                                  SEN01670
0119            CALL RESTNT(MZPI,RJ(MZPI))                                SEN01680
0120         137 CONTINUE                                                 SEN01690
0121         138   IF(NEXOP4.EQ.0) GO TO 139                            SEN01700
0122            CALL LMULT(2,DELMU,DEM,DU)                                SEN01710
0123         139   CALL GRAD(2)                                          SEN01720
0124            DF=(DF-VAL)/DEM                                           SEN01730
0125            DO 140 I=1,N                                              SEN01740
0126            DELX(I)=(DELX(I) - DELX0(I))/DEM                          SEN01750
0127         140 CONTINUE                                                 SEN01760
0128            PAR(J) = X2H(J)                                           SEN01770
                C  HAVING ALREADY FACTORED A, SOLVE A*X=B FOR X,          SEN01780
                C  WHERE B=DELX AND X=DX/DA(J).                           SEN01790
```

Figure 34.--*continued*

```
0129            CALL INVERS(2)                                                  SEN01800
                C  PRINT OUT DX/DA(J)                                           SEN01810
0130            WRITE(6,150) J                                                  SEN01820
0131          150 FORMAT(47H  X-DERIVATIVES ARE WITH RESPECT TO PARAMETER ,I2)   SEN01830
0132            DO 170 I=1,N,6                                                  SEN01840
0133            II=MIN0(I+5,N)                                                  SEN01850
0134            WRITE(6,160) (JJ,DELX(JJ), JJ=I,II)                             SEN01860
0135          160 FORMAT( 6(4H DX(,I2,2H)=,G14.7) )                             SEN01870
0136          170 CONTINUE                                                      SEN01880
                C CALCULATION OF GP VARIABLE SENSITIVITY                        SEN01890
0137            IF(NEXOP6.NE.1) GO TO 670                                       SEN01900
0138            INDEX=1                                                         SEN01910
0139            CALL TRANS(INDEX)                                               SEN01920
0140            WRITE(6,610) J                                                  SEN01930
0141          610 FORMAT(2X,'CORRESPONDING DERIVATIVES OF GP VARIABLES WRT'     SEN01940
                   1,' PARAMETER',I2,'ARE')                                     SEN01950
0142            DO 680 I=1,N,6                                                  SEN01960
0143            II=MIN0(I+5,N)                                                  SEN01970
0144            WRITE(6,160)(JJ,DELX(JJ),JJ=I,II)                               SEN01980
0145          680 CONTINUE                                                      SEN01990
0146            DO 690 I=1,N                                                    SEN02000
0147          690 DELX(I)=TDELX(I)                                              SEN02010
0148          670 CONTINUE                                                      SEN02020
0149            IF(NEXOP4.EQ.0) GO TO 375                                       SEN02030
0150            IF(M.EQ.0) GO TO 301                                            SEN02040
0151            CALL LMULT(3,DELMU,DEM,DU)                                      SEN02050
0152            WRITE(6,350) J                                                  SEN02060
0153          350   FORMAT(/  41H  U-DERIVATIVES WITH RESPECT TO PARAMETER    ,I2)  SEN02070
0154            DO 351 I=1,M,6                                                  SEN02080
0155            II = MIN0(I+5,M)                                                SEN02090
0156            WRITE(6,352) ((JJ,DU(JJ)),JJ=I,II)                              SEN02100
0157          352   FORMAT(6(4H DU(,I2,2H)=,G14.7))                           SEN02110
0158          351   CONTINUE                                                  SEN02120
0159          301   IF(MZ.EQ.0) GO TO 375                                     SEN02130
0160            CALL LMULT(4,DELMU,DEM,DU)                                      SEN02140
0161            WRITE(6,360) J                                                  SEN02150
0162          360   FORMAT(/  41H  W-DERIVATIVES WITH RESPECT TO PARAMETER    ,I2)  SEN02160
0163            DO 361 I=1,MZ,6                                                 SEN02170
0164            II = MIN0(I+5,MZ)                                               SEN02180
0165            WRITE(6,362) ((JJ,DU(JJ+M)), JJ=I,II)                           SEN02190
0166          362   FORMAT(6(4H DW(,I2,2H)=,G14.7))                           SEN02200
0167          361   CONTINUE                                                  SEN02210
0168          375   CONTINUE                                                  SEN02220
                C  COMPUTE DF/DA(J).                                            SEN02230
0169            DO 180 I=1,N                                                    SEN02240
0170            DF = DF + X3H(I)*DELX(I)                                        SEN02250
0171          180 CONTINUE                                                      SEN02260
                C  PRINT DF/DA(J).                                              SEN02270
0172            WRITE(6,190) DF                                                 SEN02280
0173          190 FORMAT(/ 10X,13HDF(X(M))/DA= ,G14.6/10H **********/)          SEN02290
0174          200 CONTINUE                                                      SEN02300
0175            GO TO 202                                                       SEN02310
0176          201 WRITE(6,106) J                                                SEN02320
0177          106 FORMAT(1H ,21HTERMINATING PARAMETER,I3,16H DUE TO DPAR = 0   /)  SEN02330
0178            GO TO 200                                                       SEN02340
0179          202 NT3=NP                                                        SEN02350
0180            CALL REJECT                                                     SEN02360
0181            DO 205 I=1,N                                                    SEN02370
0182            DELX(I) = DELTX(I)                                              SEN02380
0183          205 DELX0(I) = DELT(I)                                            SEN02390
0184          400 CONTINUE                                                      SEN02400
C185          500 RETURN                                                        SEN02410
0186            END                                                             SEN02420
```

Figure 34.--*continued*

```
FORTRAN IV G LEVEL  21                SET              DATE = 80242        12/14/52

0001                SUBROUTINE SET (TMMAX)                                  SET00040
                C                                                           SET00050
                C     FEBUARY 1971                                          SET00060
                C                                                           SET00070
                C SET STORES THE TIME AT WHICH THE PROBLEM IS BEGUN          SET00080
0002                COMMON /TSW/ NSWW                                       SET00090
0003                COMMON /TMX/ TMO,EXT,EXT90                              SET00100
                C                                                           SET00110
                C  SECOND GIVES JOB CPU EXECUTION TIME IN 1/1000 OF A SECOND SET00120
                C                                                           SET00130
0004                CALL SECOND (TMO)                                       SET00140
0005                EXT=TMMAX+TMO                                           SET00150
0006                EXT90= TMO + 0.90*TMMAX                                 SET00160
0007                NSWW=1                                                  SET00170
0008                RETURN                                                  SET00180
0009                END                                                     SET00190
```

Figure 35.--Subroutine SET.

6.28 STØRE

Subroutine STØRE stores the values of the current point x and the associated values of F, W, and G in a temporary storage area. These values are restored by a call to REJECT. The listing for subroutine STØRE appears as Figure 36.

```
FORTRAN IV G LEVEL  21              STORE              DATE = 80242        12/15/22

0001                SUBROUTINE STORE                                        STO00040
                C                                                           STO00050
                C              OCTOBER 1970                                 STO00060
                C                                                           STO00070
                C STORE STORES THE VALUES OF THE CURRENT POINT AND THE ASSOCIATED STO00080
                C VALUES OF THE FUNCTIONS IN A TEMPORARY AREA.               STO00090
0002                IMPLICIT REAL*8(A-H,O-Z)                                STO00100
0003                REAL*4 RHOIN,RATIO,EPSI,THETAO                          STO00110
0004                COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1      STO00120
0005                COMMON /EQAL/ M, M1, M2                                 STO00130
0006                COMMON/VALUE/F,G,PO,RSIGMA,RJ(90),RHO                   STO00140
0007                COMMON/CAST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO, STO00150
                   1RSIG1,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,      STO00160
                   2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),              STO00170
                   3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS             STO00180
0008                DO 10 I=1,N                                             STO00190
0009           10   X1(I)=X(I)                                              STO00200
0010                MM2=M+M2                                                STO00210
0011                DO 20 J=1,MM2                                           STO00220
0012           20   RJ1(J)=RJ(J)                                            STO00230
0013                P1=PO                                                   STO00240
0014                F1=F                                                    STO00250
0015                G1=G                                                    STO00260
0016                RSIG1=RSIGMA                                            STO00270
0017                M1=M                                                    STO00280
0018                RETURN                                                  STO00290
0019                END                                                     STO00300
```

Figure 36.--Subroutine STØRE.

6.29 TCHECK

If the print option has been set so that printouts occur only after a subproblem has been called, then TCHECK is called after each iteration of the algorithm used to minimize the W function. The elapsed time is computed as is done in TIMEC but it is not printed out. If the elapsed time exceeds 90 percent of the estimated time limit, then the print option is changed so a printout occurs after each iteration of the procedure being used to minimize the W function. If the elapsed time exceeds the user specified time limit, then the routine will cause termination of the attempt to solve the problem by setting NSWW equal to 2. Subroutine TCHECK appears as Figure 37.

```
FORTRAN IV G LEVEL  21                TCHECK              DATE = 80243          10/40/07

0001                SUBROUTINE TCHECK                                                 TCH00050
                C             FEBUARY 1971                                          TCH00060
                C                                                                   TCH00070
                C TCHECK CHECKS THE NUMBER OF SECONDS THAT HAVE ELAPSED SINCE THE START TCH00080
                C OF THE PROBLEM. IF THE SOLUTION IS TAKING LONGER THAN 90 PER-CENT  TCH00090
                C OF THE ESTIMATED MAXIMUM TIME, A SWITCH IS SET TO GIVE MORE OUTPUT. TCH00100
0002                COMMON /TSW/ NSWW                                               TCH00110
0003                COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10         TCH00120
0004                COMMON /TMX/ TMO,EXT,EXT90                                      TCH00130
0005                CALL SECOND (SECS)                                              TCH00140
0006                IF (SECS.LT.EXT90) RETURN                                       TCH00150
                C GETTING CLOSE TO EXCEEDING THE TIME LIMIT SET OUTPUT OPTION TO GIVE TCH00160
                C MORE OUTPUT.                                                      TCH00170
0007                NT3=2                                                           TCH00180
0008                X=SECS - TMO                                                    TCH00190
0009                WRITE(6,10) X                                                   TCH00200
0010            10  FORMAT (6X,5HTIME=,F9.3,8H SECONDS  )                          TCH00210
0011                IF(SECS .GT. EXT) NSWW=2                                        TCH00220
0012                CALL OUTPUT (1)                                                 TCH00230
0013                RETURN                                                          TCH00240
0014                END                                                             TCH00250
```

Figure 37.--Subroutine TCHECK.

6.30 TIMEC

Subroutine TIMEC calls the same routine as called by SET to obtain the current status of the computer's time clock. It then prints out the elapsed time since the call to SET. If the elapsed time is greater than the user specified maximum time limit, TIMEC will cause the termination of the attempt to solve the problem by setting NSWW

equal to 2. The listing for subroutine TIMEC is shown below in Figure 38.

```
FORTRAN IV G LEVEL 21                TIMEC              DATE = 80242        12/15/53

0001                 SUBROUTINE TIMEC                                          TIM00030
                C                                                              TIM00040
                C    FEBUARY 1971                                              TIM00050
                C                                                              TIM00060
                C TIMEC CHECKS THE NUMBER OF SECONDS THAT HAVE ELAPSED SINCE THE START  TIM00070
                C OF THE PROBLEM. IT PRINTS THIS NUMBER. IF THE SOLUTION IS TAKING      TIM00080
                C LONGER THAN THE ESTIMATED MAXIMUM TIME, A SWITCH IS SET TO TERMINATE  TIM00090
                C THE RUN.                                                     TIM00100
0002                 COMMON /TSW/ NSWW                                         TIM00110
0003                 COMMON /TMX/ TMO,EXT,EXT90                                TIM00120
                C                                                              TIM00130
                C  SECOND GIVES JOB CPU EXECUTION TIME IN 1/1000 OF A SECOND   TIM00140
                C                                                              TIM00150
0004                 CALL SECOND (SECS)                                        TIM00160
0005                 X=SECS-TMO                                                TIM00170
0006                 WRITE (6,20) X                                            TIM00180
0007                 IF (SECS.LT.EXT) GO TO 10                                 TIM00190
0008                 NSWW=2                                                    TIM00200
0009            10   RETURN                                                    TIM00210
                C                                                              TIM00220
0010            20   FORMAT (6X,5HTIME=F9.3,8H SECONDS)                        TIM00230
0011                 END                                                       TIM00240
```

Figure 38.--Subroutine TIMEC.

6.31 TRANS

TRANS is a special purpose subroutine that performs an exponential transformation of the solution vector and sensitivity results. This subroutine is useful when solving problem $c(x)$, the convex equivalent of the geometric programming problem $G(t)$. The former problem is obtained from the latter via the transformation $t = e^{-x_i}$, $i=1,\ldots,n$. TRANS is used to transform the results to the original space of the variable t. The motivation for coding this subroutine was the computational difficulty experienced by some users while solving geometric programming problems with SUMT. TRANS is called by the subroutines ØUTPUT and SENS. Figure 39 is a listing of subroutine TRANS.

6.32 XMØVE

Subroutine XMØVE contains most of the logic of the algorithm used to minimize the W function for a given value of r (RHØ). First, a

```
FORTRAN IV G LEVEL 21          TRANS          DATE = 80243          12/19/30

0001          SUBROUTINE TRANS(INDEX)                                    TRA00050
          C SUBROUTINE TRANS CODED BY GHAEMI(1979) TRANSFORMS THE OPTIMAL  TRA00060
          CX* AND SENSITIVITY RESULTSTO T SPACE WHERE                     TRA00070
          C T=EXP(-X). THIS ROUTINE IS USEFULL WHEN USER SOLVES THE CONVEX TRA00080
          C EQUIVALLENT OF GP PROBLEMS OBTAINED BY THE ABOVE EXPONENTIAL   TRA00090
          C TRANSFORMATION.                                               TRA00100
0002          IMPLICIT REAL*8(A-H,O-Z)                                    TRA00110
0003          COMMON/SHAPE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1          TRA00120
0004          COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,     TRA00130
             1RS1ol,G1,X1(45),X2(45),X3(45),XR2(45),XR1(45),PR1,          TRA00140
             2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                  TRA00150
             3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                 TRA00160
0005          COMMON/ABG5/TX(45),TDELX(45)                                TRA00170
0006          COMMON/ABG6/XX(45)                                          TRA00180
0007          IF(INDEX.EQ.1) GO TO 20                                     TRA00190
0008          DO 10 I=1,N                                                 TRA00200
0009          TX(I)=X(I)                                                  TRA00210
0010          X(I)=DEXP(-X(I))                                            TRA00220
0011       10 XX(I)=X(I)                                                  TRA00230
0012          RETURN                                                      TRA00240
0013       20 DO 30 I=1,N                                                 TRA00250
0014          TDELX(I)=DELX(I)                                            TRA00260
0015       30 DELX(I)=-DEXP(-X(I))*DELX(I)                                TRA00270
0016          RETURN                                                      TRA00280
0017          END                                                         TRA00290
```

Figure 39.--Subroutine TRANS.

direction is chosen in which it is believed the W function will initially decrease. Then ØPT is used to find a minimum of the W function in that direction. Having generated a new point giving a lower value of the W function, XMØVE returns control to BØDY.

Subroutine XMØVE contains three procedures for generating a direction vector. The choice of the procedure used is controlled by experimental option 2 (NEXØP2). If NEXØP2 = 1, then Newton's method is used to choose the direction vector S, that $S^i = -[\nabla^2 W(x^i, r_k)]^{-1} \nabla W(x^i)$. If $\nabla^2 W(x^i, r^k)$ is not a positive definite matrix, S^i is generated by rules outlined on page 167 in Fiacco and McCormick [10].

If NEXØP2 = 2, then the negative of the gradient is used as the direction vector, $S^i = -\nabla W(x^i, r^k)$.

When NEXØP2 = 3 a variable matrix method is used to generate S^i. This method is McCormick's modification of the Fletcher-Power-Davidon, as described on pages 170-175 in Fiacco and McCormick [10]. Subroutine XMØVE is shown as Figure 40.

```
FORTRAN IV G LEVEL 21                XMOVE              DATE = 80242        12/16/26

0001                  SUBROUTINE XMOVE                                                XM000040
                C                                                                     XM000050
                C            MARCH 1971                                               XM000060
                C                                                                     XM000070
                C XMOVE DETERMINES THE VECTOR ALONG WHICH THE SEARCH FOR A MINIMUM IS  XM000080
                C USING OPT.                                                          XM000090
0002                  IMPLICIT REAL*8(A-H,O-Z)                                        XM000100
0003                  REAL*4 RHOIN,RATIO,EPS1,THETAO,XEP1,XEP2                        XM000110
0004                  COMMON/SHARE/X(45),DEL(45),A(45,45),N,M,MN,NP1,NM1              XM000120
0005                  COMMON/CRST/DELX(45),DELXO(45),RHOIN,RATIO,EPSI,THETAO,         XM000130
                     1RSIG1,G1,XL(45),X2(45),X3(45),XR2(45),XR1(45),PR1,              XM000140
                     2PR2,P1,F1,RJ1(90),DOTT,PGRAD(45),DIAG(45),                      XM000150
                     3 PREV3,ADELX, NTCTR, NUMINI, NPHASE, NSATIS                     XM000160
0006                  COMMON/EXPOPT/NEXOP1,NEXOP2,NEXOP3,NEXOP4,NEXOP5,XFP1,XEP2      XM000170
0007                  COMMON/XVE/SIG(45),YY(45),XXX(45),DELL(45)                      XM000180
                C--NEXOP2 DETERMINES HOW MOVE IS TO BE MADE                           XM000190
                C NEXOP2 = 1  USE MODIFIED  NEWTON RAPHSON METHOD.                    XM000200
                C        = 2  USE MODIFIED  NEWTON RAPHSON METHOD, BUT ADD DELXO TO   XM000210
                C             ORTHOGONAL MOVE VECTOR IF HESSIAN IS INDEFINITE.        XM000220
                C        = 3  USE STEEPEST DESCENT METHOD.                            XM000230
                C        = 4  USE MCCORMICK,S MODIFICATION OF THE FLETCHER-POWELL      XM000240
                C             METHOD.                                                 XM000250
0008                  GO TO (10,10,180,30), NEXOP2                                    XM000260
                C--NEWTON -RAPH WITH WHATEVER METHOD IS IN INVERSE                     XM000270
0009            10    CALL GRAD (1)                                                   XM000280
                C--ONE (1) MEANS ACCUMULATE MATRIX OF SECOND PARTIAL DERIVATIVES       XM000290
0010                  CALL SECORD (1)                                                 XM000300
0011                  DO 20 I=1,N                                                     XM000310
0012            20    DELX(I)=DELXO(I)                                                XM000320
0013                  CALL INVERS (1)                                                 XM000330
                C IF A NONPOSITVE PIVOT IS ENCOUNTERED IN INVERSE AN ATTEMPT IS MADE TO XM000340
                C COMPUTE A VECTOR HAVING A POSITIVE DOT PRODUCT WITH A NEGATIVE       XM000350
                C EIGENVECTOR AND THE NEGATIVE OF DEL P.                              XM000360
0014                  CALL STORE                                                      XM000370
0015                  CALL OPT                                                        XM000380
0016                  RETURN                                                          XM000390
                C--F-P-D-MCC MOVE                                                     XM000400
0017            30    CALL GRAD (2)                                                   XM000410
                C--MN IS NO. OF MOVES FOR THIS VALUE OF RHO                           XM000420
0018                  IF (MN.NE.0) GO TO 70                                           XM000430
0019            40    IREP=0                                                          XM000440
0020                  IT=0                                                            XM000450
                C--SET INITIAL GUESS INVERS MATRIX OF SECOND PARTIAL DERIVATIVES       XM000460
                C-- USE PARTIAL INVERSE IF KNOWN                                      XM000470
0021                  DO 50 I=1,N                                                     XM000480
0022                  DO 50 J=1,N                                                     XM000490
0023            50    A(I,J)=0.0                                                      XM000500
0024                  DO 60 I=1,N                                                     XM000510
0025            60    A(I,I)=1.0                                                      XM000520
0026            70    DO 80 I=1,N                                                     XM000530
0027            80    DELX(I)=DELXO(I)                                                XM000540
0028                  IF (IREP.GT.N) GO TO 40                                         XM000550
0029                  IF (IT.EQ.0) GO TO 130                                          XM000560
0030                  DO 90 I=1,N                                                     XM000570
0031                  SIG(I)=X(I)-XXX(I)                                              XM000580
0032            90    YY(I)=DELL(I)-DELXO(I)                                          XM000590
                C--NEGATIVE GRADIENT STORED  AND COMPUTED                             XM000600
                C--COMPUTE  HY                                                        XM000610
0033                  DO 100 I=1,N                                                    XM000620
0034                  DELX(I)=0.0                                                     XM000630
0035                  DO 100 J=1,N                                                    XM000640
0036            100   DELX(I)=DELX(I)+A(I,J)*YY(J)                                    XM000650
                C--COMPUTE  Y(SIG -HY)-1                                              XM000660
0037                  ZCON=0.0                                                        XM000670
0038                  DO 110 I=1,N                                                    XM000680
0039            110   ZCON=ZCON+YY(I)*(SIG(I)-DELX(I))                                XM000690
0040                  IF (ZCON.EQ.0.0) GO TO 130                                      XM000700
0041                  IREP=IREP+1                                                     XM000710
0042                  ZC=1./ZCON                                                      XM000720
                C-- UPDATE   H MATRIX USING  MCC FORMULA WHEN SCALAR NE 0             XM000730
0043                  DO 120 I=1,N                                                    XM000740
0044                  T1=ZC*(SIG(I)-DELX(I))                                          XM000750
0045                  DO 120 J=1,N                                                    XM000760
0046                  A(I,J)=A(I,J)+T1*(-DELX(J)+SIG(J))                              XM000770
0047            120   A(J,I)=A(I,J)                                                   XM000780
                C-- STORE CURRENT POINT AND CURRENT GRADIENT (NEG)                    XM000790
0048            130   DO 140 I=1,N                                                    XM000800
0049                  XXX(I)=X(I)                                                     XM000810
0050            140   DELL(I)=DELXO(I)                                                XM000820
0051                  DO 150 I=1,N                                                    XM000830
0052                  DELX(I)=0.0                                                     XM000840
0053                  DO 150 J=1,N                                                    XM000850
0054            150   DELX(I)=DELX(I)+A(I,J)*DELXO(J)                                 XM000860
0055                  ZC1=0.0                                                         XM000870
0056                  DO 160 I=1,N                                                    XM000880
0057            160   ZC1=DELX(I)**2+ZC1                                              XM000890
0058                  ZC1=DSQRT(ZC1)                                                  XM000900
```

Figure 40.--Subroutine XMØVE.

```
0059            DO 170 I=1,N                                        XM000910
0060      170   DELX(I)=DELX(I)/ZC1                                 XM000920
0061            CALL STORE                                          XM000930
0062            CALL OPT                                            XM000940
0063            IT=IT+1                                             XM000950
0064            RETURN                                              XM000960
0065      180   CONTINUE                                            XM000970
                C STEEPEST DESCENT                                  XM000980
0066            CALL GRAD (2)                                       XM000990
0067            DO 190 I=1,N                                        XM001000
0068      190   DELX(I)=DELX0(I)                                    XM001010
0069            CALL STORE                                          XM001020
0070            CALL OPT                                            XM001030
0071            RETURN                                              XM001040
0072            END                                                 XM001050
```

Figure 40.--*continued*

REFERENCES

[1] ARMACOST, R. L. (1976). Sensitivity analysis in parametric nonlinear programming. D.Sc. dissertation, The George Washington University.

[2] ARMACOST, R. L. and A. V. FIACCO (1974). Computational experience in sensitivity analysis for nonlinear programming. *Math. Programming* 6 (3), 301-325.

[3] ARMACOST, R. L. and A. V. FIACCO (1975). Second-order sensitivity analysis in NLP and estimates by penalty function methods. Technical Paper Serial T-324, Institute for Management Science and Engineering, The George Washington University.

[4] ARMACOST, R. L. and A. V. FIACCO (1976). NLP sensitivity analysis for RHS perturbations: A brief survey and recent second-order extensions. Technical Paper Serial T-334, Institute for Management Science and Engineering, The George Washington University.

[5] ARMACOST, R. L. and A. V. FIACCO (1977). Exact sensitivity analysis using augmented Lagrangians. Technical Paper Serial T-349, Institute for Management Science and Engineering, The George Washington University.

[6] ARMACOST, R. L. and A. V. FIACCO (1978). Sensitivity analysis for parametric nonlinear programming using penalty methods. *Computers and Mathematical Programming*, National Bureau of Standards Special Publication 502, 261-269.

[7] ARMACOST, R. L. and W. C. MYLANDER (1973). A guide to a SUMT-version 4 computer subroutine for implementing sensitivity analysis in nonlinear programming. Technical Paper Serial T-287, Institute for Management Science and Engineering, The George Washington University.

[8] CAUSEY, B. (1970). A method for sensitivity analysis nonlinear programming. Unpublished manuscript.

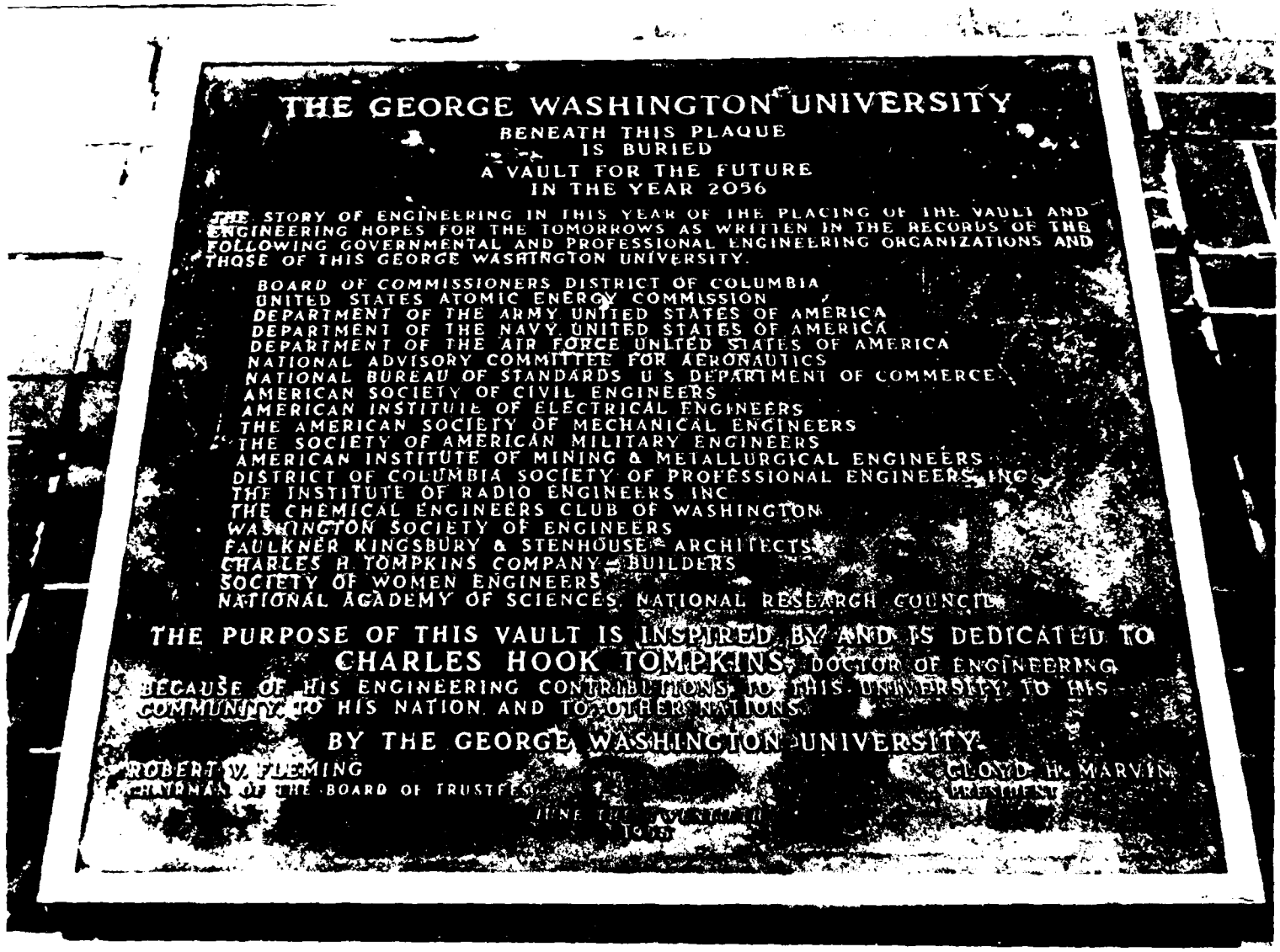
[9] FIACCO, A. V. (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Math. Programming*, 10 (3), 287-311.

[10] FIACCO, A. V. and G. P. McCORMICK (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Technique*. John Wiley and Sons, New York.

[11] GHAEMI, A. (1980). Computable stability analysis techniques for nonlinear programming: Sensitivities, optimal value bounds, and applications. D.Sc. dissertation, The George Washington University.

[12] MYLANDER, W. C. (1971). Estimating the sensitivity of a solution of a nonlinear program. Unpublished manuscript.

[13] MYLANDER, W. C., R. L. HOLMES, and G. P. McCORMICK (1971). A guide to SUMT-version 4: The computer program implementing the sequential unconstrained minimization technique for nonlinear programming. Technical Paper RAC-P-63, Research Analysis Corporation, McLean, Virginia.



To cope with the expanding technology, our society must be assured of a continuing supply of rigorously trained and educated engineers. The School of Engineering and Applied Science is completely committed to this objective.